



**Paulo Jorge  
Ferreira Antunes  
da Silva**

**Aplicação de Controlo Distribuído numa  
Calandra de Accionamento Hidráulico**

“ A melhor forma de prever  
o futuro é criá-lo.”

—Peter Druker





**Paulo Jorge  
Ferreira Antunes  
da Silva**

**Aplicação de Controlo Distribuído numa  
Calandra de Accionamento Hidráulico**

dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica do Dr. António Pereira de Melo, Professor Catedrático do Departamento de Electrónica Telecomunicações e Informática da Universidade de Aveiro



**o júri**

presidente

**José Alberto Gouveia Fonseca**

Professor Associado da Universidade de Aveiro

vogais

**António Ferreira Pereira de Melo**

Professor Catedrático da Universidade de Aveiro (orientador)

**Vítor Manuel Ferreira Santos**

Professor Associado da Universidade Aveiro



## **agradecimentos**

Ao Prof. Doutor António Pereira de Melo

Ao Eng. António da Costa Vidal

A todos os colegas que trabalham na Marcovil,S.A.

A Elisabete Gomes Cotrim Silva

Ao Tomás Daniel Cotrim Silva

À Marcovil,S.A.

À Universidade de Aveiro





## Resumo

A calandra hidráulica é uma máquina industrial concebida para conformação de chapa. Nos dias de hoje é uma máquina muito comum na indústria metalomecânica, para fabrico de caldeiras e torres eólicas. Neste trabalho foi desenvolvido um sistema de comando numérico computadorizado (CNC) com objectivo de modernizar o sistema de controlo da calandra de accionamento hidráulico.

Para implementação da solução final foi necessário definir uma base de hardware e uma base de software para controlo do processo. No sistema desenvolvido foi implementado uma rede suportada no protocolo RS485 de autómatos programáveis (denominados de PLCs, do inglês "Programmable Logic Controllers") baseada numa arquitectura mestre escravos (master and slaves) que estão directamente ligados aos actuadores e sensores usados na estrutura, no topo da pirâmide de controlo encontra-se um computador industrial que têm com funções principais o supervisionamento do processo de controlo, calculo numérico e interface com o utilizador.

O sistema de comando desenvolvido têm dois modos de trabalho, um modo automático e outro o modo manual. No modo manual a máquina é accionada directamente pelos botões e joystiks do painel de comando, sendo sempre possível ao operário basear-se no sistema de sete eixos de controlo implementados na máquina. No modo automático a máquina é autónoma e executa um conjunto de tarefas definidas pelo programa de dados. A aplicação de software principal desenvolvida para controlo da máquina designada de CNC\_CE é uma interface gráfica composta por cinco *menus* principais: Dados, Simulação, Ficheiro, Teach In e Automático.



# Conteúdo

<b>Lista de Códigos</b>	<b>v</b>
<b>1 INTRODUÇÃO</b>	<b>1</b>
1.1 Enquadramento . . . . .	1
1.2 Objectivos . . . . .	3
1.3 Estrutura da tese . . . . .	3
<b>2 CALANDRA DE ACCIONAMENTO HIDRÁULICO</b>	<b>5</b>
2.1 Descrição de Calandra de Accionamento Hidráulico . . . . .	5
2.2 Dimensionamento da máquina . . . . .	6
2.3 Modelos de Calandras de Accionamento Hidráulico . . . . .	8
2.4 Sistema Hidráulico . . . . .	9
2.4.1 Sistema de Posicionamento . . . . .	9
2.4.2 Sistema de Rotação . . . . .	10
2.5 Sistema Eléctrico . . . . .	12
<b>3 PROCESSO DE CALANDRAGEM</b>	<b>13</b>
3.1 Processo de Calandragem Tradicional . . . . .	14
3.2 Processo de Calandragem Desenvolvido . . . . .	15
<b>4 ALGORITMO MATEMÁTICO DE POSICIONAMENTO</b>	<b>21</b>
4.1 Primeiro Algoritmo de Posicionamento . . . . .	21
4.2 Segundo Algoritmo de Posicionamento . . . . .	24
<b>5 AUTOMATIZAÇÃO DA CALANDRA HIDRÁULICA</b>	<b>29</b>
5.1 Implementação dos Eixos de Controlo . . . . .	30
5.1.1 Sinais de Eixos . . . . .	30
5.2 Definição de Sinais <i>Input/Output</i> . . . . .	31

5.2.1	Sinais de <i>Input</i> . . . . .	31
5.2.2	Sinais de <i>Output</i> . . . . .	34
5.3	Estrutura <i>Hardware</i> Existente . . . . .	37
5.3.1	<i>Hardware</i> do 1º Sistema de Comando . . . . .	38
5.3.2	<i>Hardware</i> do Sistema de Comando Desenvolvido em 2005/06 . . . . .	40
5.4	Estrutura de <i>Hardware</i> do Sistema de Comando Distribuindo . . . . .	41
5.4.1	Estrutura de <i>Hardware</i> do Sistema de Comando CNC . . . . .	42
5.4.2	Estrutura de <i>Hardware</i> do Sistema de Comando NC . . . . .	46
<b>6</b>	<b><i>SOFTWARE CNC</i></b>	<b>47</b>
6.1	Ferramentas de <i>Software</i> Utilizadas . . . . .	50
6.2	Programa Principal <i>CNC_CE.exe</i> . . . . .	50
6.2.1	Diagramas de Fluxo . . . . .	53
6.2.2	Classes do Sistema . . . . .	60
6.2.3	Estruturas Principais do Código <i>CNC_CE.exe</i> . . . . .	60
6.2.4	Inicialização das Estruturas e Variáveis do Programa . . . . .	65
6.2.5	Funções Principais do Programa <i>CNC_CE.exe</i> . . . . .	66
6.2.6	<i>Menus</i> do programa <i>CNC_CE.exe</i> . . . . .	89
6.3	Programa <i>PLC MASTER</i> . . . . .	94
6.3.1	Diagramas de Fluxo . . . . .	94
6.4	Programa <i>PLC SLAVE</i> . . . . .	99
6.5	Trabalho Futuro . . . . .	101
	<b>Conclusões</b>	<b>103</b>
	<b>Bibliografia</b>	<b>105</b>

# Lista de Figuras

1.1	Esquemático de um sistema CNC . . . . .	2
2.1	Calandra Hidráulica 4R 6000/22 . . . . .	6
2.2	Tabela de capacidade de uma máquina 4R 3000/40 . . . . .	7
2.3	Topologia de Calandra de 4 Rolos . . . . .	8
2.4	Topologia de Calandra de 3 Rolos Piramidal . . . . .	8
2.5	Topologia de Calandra de 3 Rolos Assimétrica Esquerda . . . . .	9
2.6	Topologia de Calandra de 3 Rolos Assimétrica Direita . . . . .	9
2.7	Diagrama Hidráulico de Posicionamento Típico . . . . .	10
2.8	Diagrama Hidráulico de Rotação . . . . .	11
3.1	Topologia de três rolos e quatro rolos . . . . .	13
3.2	Dois Processos de Calandragem . . . . .	14
3.3	Fases de Calandragem Tradicional . . . . .	18
3.4	Fases de Calandragem do modelo Desenvolvido . . . . .	19
4.1	Diagrama da Calandra . . . . .	22
4.2	Rolos na Posição de Origem Referencia Zero dos Eixos . . . . .	24
4.3	Efeito de Deslocação dos Pontos de Contacto . . . . .	25
4.4	Diagrama da Calandra Modificado . . . . .	26
5.1	Diagrama Blocos Sistema Controlo . . . . .	38
5.2	Sistema Controlo Antigo . . . . .	39
5.3	Diagrama Blocos Sistema Controlo Distribuído CNC . . . . .	44
5.4	Calandra equipada com Sistema de Comando Distribuído CNC . . . . .	45
5.5	Diagrama Blocos Sistema Controlo Distribuído NC . . . . .	46
6.1	Imagem do Programa Visual Studio 2005 . . . . .	51
6.2	Imagem do programa GX IEC Developer 7.00 . . . . .	51

6.3	Diagrama Geral . . . . .	53
6.4	Diagrama Menu Novo Programa . . . . .	55
6.5	Diagrama Menu Simulação . . . . .	56
6.6	Diagrama Menu Lista de Programas . . . . .	57
6.7	Diagram TEACH IN . . . . .	58
6.8	Diagrama Modo Automático . . . . .	59
6.9	Imagem do Programa CNC MODO MANUAL . . . . .	90
6.10	Imagem do Programa CNC MODO NOVO PROGRAMA . . . . .	91
6.11	Imagem do Programa CNC MODO SIMULAÇÃO . . . . .	91
6.12	Imagem do Programa CNC MODO LISTA DE PROGRAMAS . . . . .	92
6.13	Imagem do Programa CNC MODO TEACH IN . . . . .	93
6.14	Imagem do Programa CNC MODO AUTOMÁTICO . . . . .	93
6.15	Diagrama Geral PLC Master . . . . .	95
6.16	Diagrama de Posicionamento Rolo Inferior . . . . .	96
6.17	Diagrama de Posicionamento Lateral Direito . . . . .	97
6.18	Diagrama de Posicionamento Lateral Esquerdo . . . . .	98
6.19	Diagrama de Posicionamento Rotação . . . . .	98

# Lista de Códigos

1	<i>Nomespaces Usados da .NET Compact Framework</i> . . . . .	60
2	Estrutura <i>public struct EstadoMaquina</i> . . . . .	60
3	Estrutura <i>public struct PosicaoRolos</i> . . . . .	61
4	Estrutura <i>public struct RegistosRolos</i> . . . . .	61
5	Estrutura <i>public struct RolosAutomato</i> . . . . .	62
6	Estrutura <i>public struct ControloRolo</i> . . . . .	62
7	Estrutura <i>public struct DadosMaquina</i> . . . . .	63
8	Estrutura <i>public struct ProgramaDados</i> . . . . .	63
9	Estrutura <i>public struct ControlMaquina</i> . . . . .	64
10	Estrutura <i>public struct Mensagem</i> . . . . .	64
11	Estrutura <i>Inicialização das Estruturas e Variáveis do Programa</i> . . . . .	66
12	Função <i>Função de Inicialização do Programa</i> . . . . .	67
13	Função <i>Função de SimulacaoRolos</i> . . . . .	68
14	Estrutura <i>struct ArcoCentro</i> . . . . .	69
15	Função <i>private void SimularProgramaDados()</i> . . . . .	71
16	Função <i>private void DisplayFiles(string dirName)</i> . . . . .	72
17	Função <i>private void ActualizarEncoder()</i> . . . . .	73
18	Função <i>private void Controlo_Manual()</i> . . . . .	74
19	Função <i>private void AUTOMATICO()</i> . . . . .	75
20	Função <i>CASE 0 - [Fase Inicial]</i> . . . . .	76
21	Função <i>CASE 1 - [Fase 1º Posicionamento]</i> . . . . .	78
22	Função <i>CASE 2 - [Fase aperto da chapa]</i> . . . . .	79
23	Função <i>CASE 3 - [Fase A e B da Calandragem]</i> . . . . .	80
24	Função <i>CASE 5 - [Fase Actualização Passagens]</i> . . . . .	81
25	Função <i>CASE 6 - [Fim de Calandragem]</i> . . . . .	83
26	Função <i>CASE 7 - [Fase Teach In]</i> . . . . .	86
27	Função <i>CASE 8 - [Paralelismo]</i> . . . . .	87

28	Função <i>CASE 9 - [Factor de Correção]</i> . . . . .	88
29	Função <i>private float PosicaoRoloLateral(float RaioActual)</i> . . . . .	88
30	Função <i>private void CalculoDistancia()</i> . . . . .	89



# Capítulo 1

## INTRODUÇÃO

### 1.1 Enquadramento

O controlo numérico nas máquinas ferramentas nasceu da necessidade de melhorar os índices de produção e acabamento em lotes de peças muito diversificadas, que às vezes apresentavam grande dificuldade de execução em máquinas tradicionais.

Por este motivo, entre outros, depois da Segunda Grande Guerra iniciaram-se estudos de desenvolvimento para implementar um sistema de comando computarizado numa máquina-ferramenta. Nesta nova máquina equipada com um comando de controlo numérico, o operador passa a ter a função de supervisionar o processo de maquinação. Esta concepção modifica bastante a relação tradicional do homem com a máquina-ferramenta, relação esta que estava fundada na capacidade do operador interpretar e traduzir a linguagem de um desenho e de um ciclo de maquinação, transformando-os em comandos coordenados de controlo contínuo de modo a determinar a construção da peça.

No novo sistema de comando da máquina, o desenho e o ciclo de maquinação são inseridos após serem traduzidos em dados numéricos e armazenados em fita de papel; esses dados, manipulados adequadamente, determinarão as operações necessárias para a produção do objecto.

Após a aplicação dos primeiros comandos de controlo numérico assistimos a uma evolução contínua que ainda está em curso, principalmente nas duas partes fundamentais: a mecânica e a electrónica de comando.

A secção mecânica é formada por partes estáticas e dinâmicas cada vez mais precisas, com uma evolução que determina uma diminuição dos atritos e das folgas permitindo assim um incremento de precisão.

A secção electrónica ou secção de comando definida como unidade de comando, é con-

stituída normalmente por uma unidade de processamento e por blocos de entradas e saídas. No caso mais típico é usado uma unidade de interface com o utilizador, onde este introduz o conjunto de dados que permitem que a máquina realize uma determinada tarefa. A este conjunto de dados designamos por programa. No seguimento do processo de execução, os dados introduzidos são analisado por uma unidade de processamento e se não forem identificados erros de execução a unidade de processamento irá dar ordens aos actuadores de forma a cumprirem todas as tarefas definidas pelo programa introduzido.

A primeira máquina com controlo numérico (NC) foi demonstrada com sucesso nos anos 50 no Massachusetts Institute of Technology (MIT).

O controlo numérico é uma forma de automação programável no qual o equipamento de processamento é controlado por programas que permitem executar sequências de operações complexas dentro da máquina, sem o auxílio do operador humano.

Actualmente, as unidades de comando utilizadas são de dois tipos: controlo numérico (NC) convencional e as de controlo numérico computadorizado (CNC). A divisão nas unidades de comando determinou uma consequente classificação em dois grupos de máquinas normalmente definidas como NC e CNC. É importante referenciar a tendência de substituição das máquinas com controlo numérico (NC) pelas máquinas de controlo numérico computadorizado (CNC).

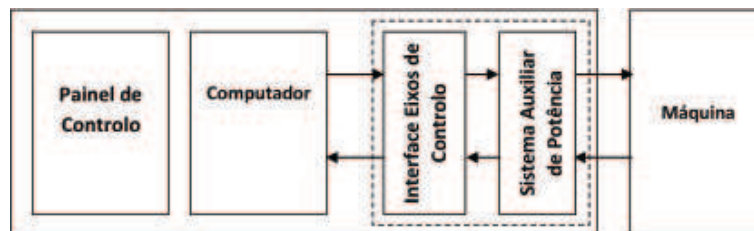


Figura 1.1: Esquematico de um sistema CNC

A evolução dos sistemas de comando nas máquinas ferramentas foi alargada às restantes máquinas da área metalomecânica. A evolução dos sistemas de comando não é sentida ao mesmo nível em toda a gama de máquinas, verificando-se um grande desenvolvimento em máquinas de corte, dobragem e furação. A Calandra é uma das máquinas que pertence ao grupo da conformação de chapa, por não ser uma máquina muito comum na área metalomecânica o seu sistema de comando comparado com outras máquinas (ex. tornos e fresas) verifica um nível de evolução e performance muito inferior.

Com o objectivo de melhorar o sistema de comando da calandra de accionamento hidráulico, neste trabalho irei apresentar o desenvolvimento de um sistema de controlo numérico computadorizado, contribuindo assim para a aproximação da performance de uma

calandra a outra máquina do ramo da metalomecânica.

Nos últimos anos verificou-se um acréscimo do mercado das calandras accionamento hidráulicas, por estas estarem directamente relacionadas com os recentes mercado de produção e transporte de energia. O forte investimento na construção de estruturas para a produção de energia eólica(fabricação de torres eólicas), produção de álcool(fabricação de caldeiras) e para construção de linhas de transporte de energia(construção de pipeline e gasodutos), foram factores determinantes para o desenvolvimento do sistema de comando das calandras hidráulicas.

Este trabalho surge na continuação do projecto de final de curso da licenciatura em Engenharia Electrónica e Telecomunicações do ano lectivo 2005/06, onde se iniciou o desenvolvimento de uma estrutura de controlo para aplicar a uma calandra de accionamento hidráulico. Na primeira estrutura de comando implementada verificou-se algumas dificuldades em controlar o processo, contudo o trabalho desenvolvido foi importante para o início dos projectos que se seguiram.

## 1.2 Objectivos

No seguimento do conteúdo do ponto anterior este trabalho têm como objectivo desenvolver um sistema de controlo numérico computarizado (CNC) para uma calandra de accionamento hidráulico. O sistema de controlo terá uma topologia de controlo distribuído, onde iremos ter um supervisor do processo geral e controladores de processos locais. O supervisor será composto por um PC Industrial que ira supervisionar uma rede de autómatos programáveis (PLC's). O desenvolvimento do sistema de controlo numérico têm como objectivo construir um sistema de acesso simples para o utilizador (tipo "user frendly") e que do mesmo modo permita um controlo total do processo de calandragem.

## 1.3 Estrutura da tese

Esta tese é composta por um total de seis capítulos, o capitulo 1 - Introdução é feito uma breve apresentação da evolução dos sistemas de controlo numérico e dos objectivos a que se propõe o trabalho. No capitulo 2 é feita a apresentação da calandra de accionamento hidráulico e uma analise do sistema hidráulico e eléctrico da maquina. No capitulo 3 é feito o dimensionamento e identificação de todos os sinais de entrada/saída a implementar no sistema de controlo bem como se define os eixos de controlo e a estrutura de hardware a usar na construção do comando. Os capítulos 4 e 5 são reservados para o estudo do

processo de calandragem e a sua melhor adaptação para o sistema de controlo numérico computarizado (CNC), é também definido um novo algoritmo de posicionamento para o processo de calandragem. No ultimo capítulo é apresentado o software desenvolvido e as suas potencialidades.

Em anexo encontra-se o manual de utilizador do sistema desenvolvido, esquemas hidráulicos e eléctricos do sistema final.

## Capítulo 2

# CALANDRA DE ACCIONAMENTO HIDRÁULICO

A calandra de accionamento hidráulico mantém os conceitos básicos de construção, como os rolos de aço e a geometria piramidal da calandra de accionamento mecânico. Na calandra de accionamento hidráulico, o movimento de posicionamento dos rolos passou a ser accionado por cilindros hidráulicos, em vez dos antigos veios roscados e manivelas das calandras accionamento mecânico.

### 2.1 Descrição de Calandra de Accionamento Hidráulico

A calandra de accionamento mecânico era constituída por três rolos de aço horizontais dispostos de uma forma piramidal. O posicionamento dos rolos laterais era feito mecânicamente por veios roscados e manivelas, a rotação era accionada electricamente por meio de engrenagens acopladas a um motor eléctrico.

Com o passar dos tempos e a necessidade para maiores capacidades de calandragem, foi obrigatória uma evolução no sistema de accionamento da calandra. A introdução do accionamento hidráulico da calandra veio-nos permitir um aumento da capacidade de calandragem e um maior ganho na potência útil.

A Calandra de accionamento hidráulico mais comum é composta por quatro rolos de aço, onde três rolos são moveis e um fixo. O rolo inferior e laterais são movimentados no sentido do curso de deslocamento por dois cilindros hidráulicos colocados nas extremidades de cada um dos rolos. Os quatro rolos têm um movimento de rotação accionado por um grupo de redutor e motor hidráulico colocados numa das extremidades de cada rolo.



Figura 2.1: Calandra Hidráulica 4R 6000/22

## 2.2 Dimensionamento da máquina

O dimensionamento do sistema hidráulico é feito em função do limite máximo de calandragem. Este limite é determinado pela conjugação de vários factores: o menor diâmetro de calandragem, largura máxima, espessura máxima e resistência mecânica da chapa (tipicamente  $370 \text{ N/mm}^2$ , chapa aço carbono ST 37).

Os parâmetros principais que interferem no dimensionamento da calandra são:

- (a) Largura da Chapa
- (b) Espessura da Chapa
- (c) Resistência Mecânica
- (d) Diâmetro Mínimo de Calandragem

A capacidade da máquina varia em função dos factores referidos, sendo calculada para o limite máximo de capacidade do menor diâmetro possível para a espessura e largura nominal da máquina. Alterando os factores da largura, espessura e diâmetro teremos

um numero indeterminado de possibilidades de capacidade de calandragem para a mesma máquina. Com a alteração dos factores determinantes para dimensionamento da maquina iremos encontrar conjugações equivalentes de capacidade para a mesma maquina.

Na Figura 2.2 apresento a tabela da capacidade de calandragem de uma maquina 4R 3000/40, quatro rolos(4R) largura de 3000mm(3000) e espessura nominal de 40 mm(40) para chapa aço carbono ST 37.

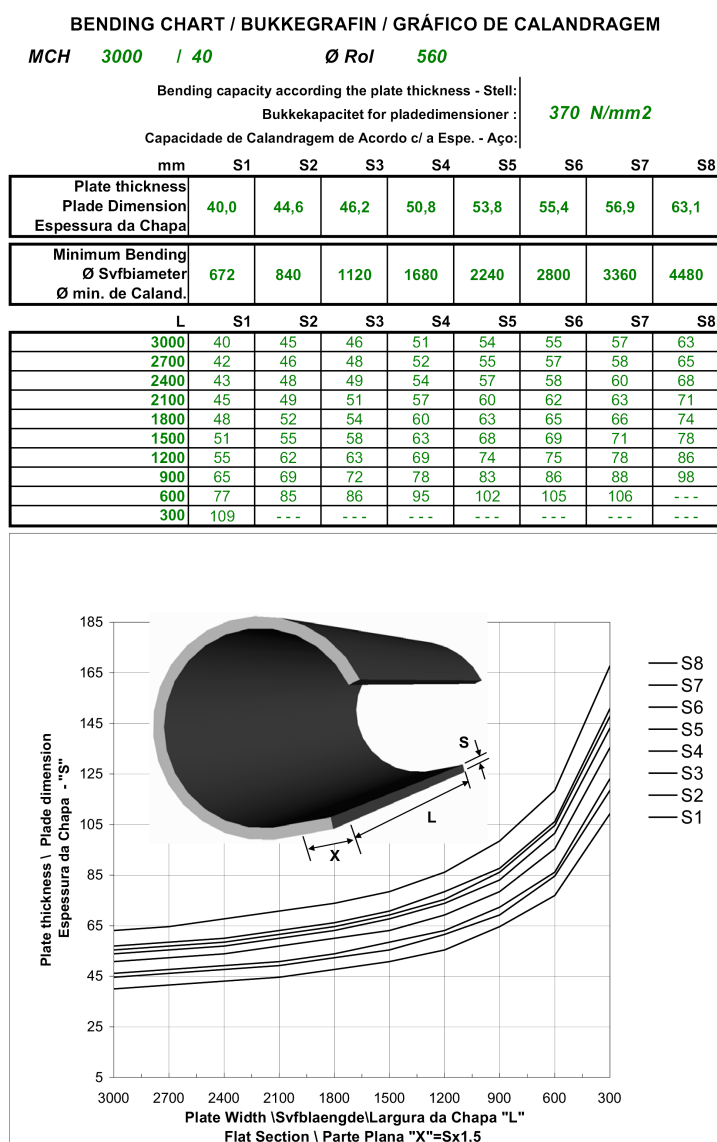


Figura 2.2: Tabela de capacidade de uma máquina 4R 3000/40

O dimensionamento físico é feito em função da força necessária para a deformação da chapa, após determinar a força é calculado o diâmetro dos rolos da máquina de forma a sustentarem a força exercida pela chapa no momento da calandragem.

O Sistema hidráulico é dimensionado em função dos quatro parâmetros já enumerados,

determinado a força necessária para a deformação, é dimensionado a superfície dos vários cilindros hidráulicos de suporte das extremidades dos rolos da calandra. Normalmente o cálculo do dimensionamento é em função de uma pressão de trabalho máxima de 200 bar nas máquinas construídas pela empresa onde desenvolvo este trabalho. Outro factor importante no dimensionamento da estrutura da calandra é o binário necessário para accionar os rolos da calandra. Com o calculo do binário máximo é determinado o conjunto de redutor, motor e bomba hidráulica para cada rolo.

## 2.3 Modelos de Calandras de Accionamento Hidráulico

Actualmente existem duas geometrias de calandras hidráulicas, uma composta por 3 rolos de aço outra composta por 4 rolos de aço.

A máquina de 3 rolos existe com duas configurações possíveis, uma de forma piramidal e outra de forma assimétrica, contudo a calandra de 4 rolos é mais usual nos dias de hoje sendo esta equivalente a uma sobreposição de duas máquinas assimétrica de 3 rolos, uma direita e outra esquerda.

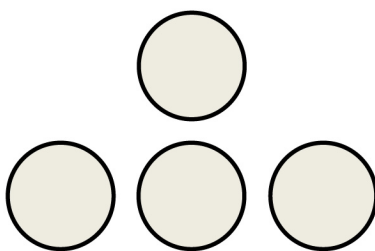


Figura 2.3: Topologia de Calandra de 4 Rolos

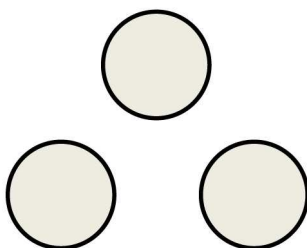


Figura 2.4: Topologia de Calandra de 3 Rolos Piramidal

A grande vantagem da máquina de 4 rolos, é a sua perfeição de acabamento das formas cilíndricas, devido à existência de mais um rolo permite que as duas extremidades da chapa fiquem com menor área secção recta, a rapidez de execução dos trabalhos é muito maior,



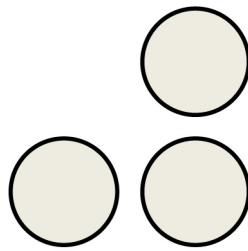


Figura 2.5: Topologia de Calandra de 3 Rolos Assimétrica Esquerda

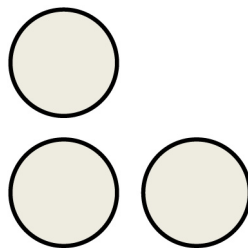


Figura 2.6: Topologia de Calandra de 3 Rolos Assimétrica Direita

mas a grande diferença está na sua estrutura de quatro rolos, permitir implementar um eixo de controlo de rotação cuja posição é possível controlar numericamente.

## 2.4 Sistema Hidráulico

Nas máquinas onde estou a realizar este trabalho, o sistema hidráulico é composto por dois circuitos distintos. Um circuito para o sistema de posicionamento e outro para o sistema de rotação.

### 2.4.1 Sistema de Posicionamento

O sistema hidráulico de uma calandra é composto por seis cilindros hidráulicos principais e um secundário. Os cilindros hidráulicos principais estão distribuídos dois a dois pelas extremidades dos rolos laterais e do rolo inferior, existe outro cilindro hidráulico que acciona o suporte da extremidade do rolo superior.

O sistema é dimensionado de forma a ter uma velocidade de posicionamento de 500mm/min na sua velocidade rápida, para além desta velocidade existe a possibilidade de uma velocidade de posicionamento lento de forma que o sistema permita um erro de posicionamento de 0,1mm.

O circuito hidráulico é composto por um agregado de quatro bombas hidráulicas de



## 2.4.2 Sistema de Rotação

O Sistema de accionamento dos rolos é constituído por redutores planetários e motores hidráulicos orbitais e pistons radiais, que estão acoplados nas extremidade dos rolos da calandra. O conjunto do redutor e motor têm a função transferir o binário necessário para fazer rodar os rolos da calandra. O dimensionamento destes pares redutor e motor é função do binário necessário para satisfazer a binário resistente que a chapa exerce ao movimento de rotação dos rolos da calandra.

O circuito hidráulico da rotação é composto normalmente por um agregado de três bombas de engrenagens hidráulicas, uma dedicada ao circuito do rolo superior, outra o circuito do rolo inferior e a terceira dedicada os dois rolos laterais.

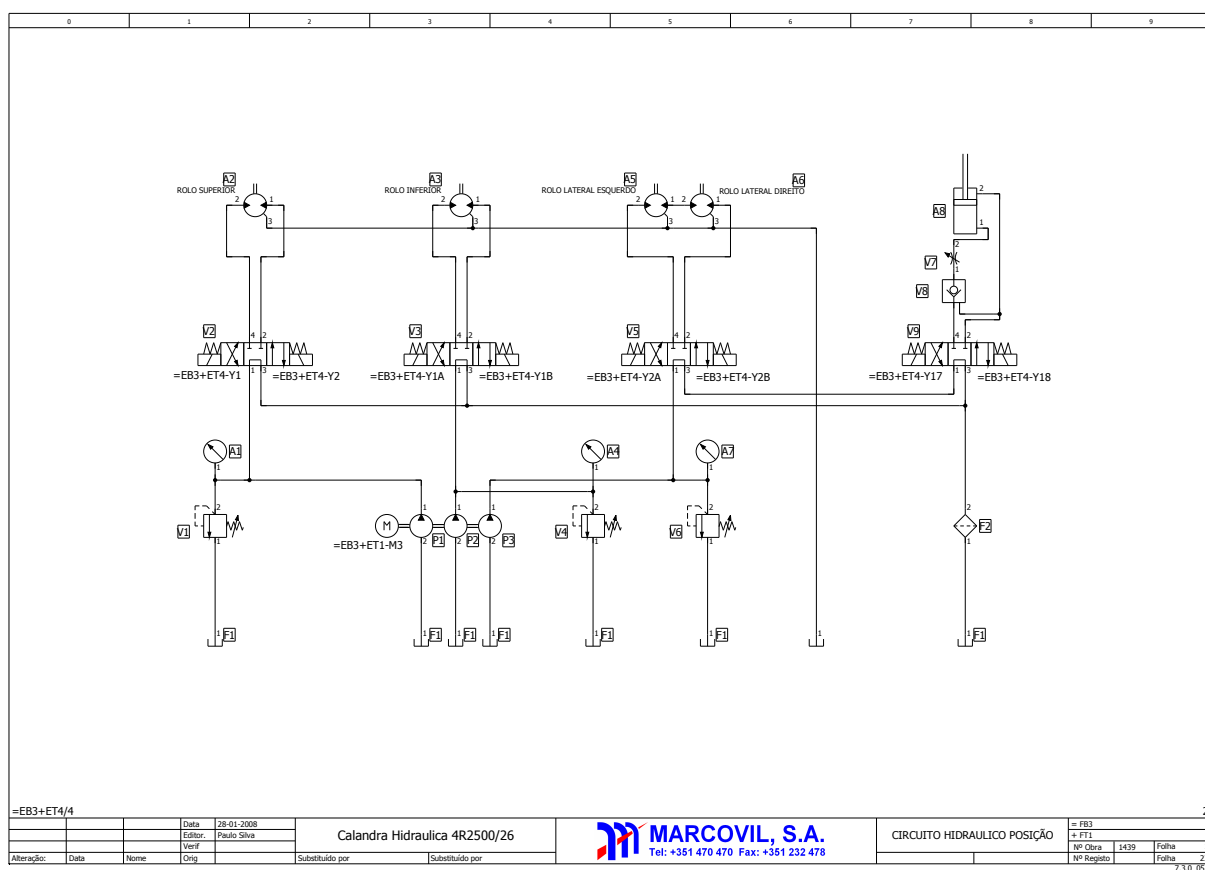


Figura 2.8: Diagrama Hidráulico de Rotação

Um dos parâmetros fundamentais para o bom funcionamento de uma calandra é a velocidade dos seus rolos. Normalmente o diâmetro do rolo superior e inferior é diferente do diâmetro dos rolos laterais, este facto deve-se á maior concentração de carga se localizar nos dois rolos principais no acto da calandragem.

Para que todos os rolos tenham uma velocidade periférica de rotação aproximada é necessário assegurar que a relação composta pelo diâmetro do rolo, relação de transmissão do redutor, cilindrada do motor hidráulico e cilindrada da bomba hidráulica conjuguem a mesma velocidade periférica em todos os rolos.

São usadas também electroválvulas direccionais no circuito de rotação para permitir definirmos o sentido de rotação de cada rolo, sentido directo ou inverso.

## 2.5 Sistema Eléctrico

O sistema eléctrico da calandra é composto por normalmente por dois motores eléctricos trifásicos de quatro pólos, responsáveis pelo accionamento de dois agregados de bombas hidráulicas, o motor que acciona o grupo hidráulico da rotação normalmente têm uma potência superior. O motor dedicado ao agregado de bombas do circuito de rotação, está controlado por um variador de frequência que nos permite um controlo de potência e variação da velocidade de rotação.

O sistema de accionamento hidráulico é feito por intermédio de um grupo de electroválvulas direccionais de quatro vias três posições, que são comutadas por um interface de relés.

O sistema eléctrico é também composto por um circuito de segurança de grau quatro, com fins de curso de emergência e betoneira de emergência, está também implementado um rele de detecção sequência de fases para proteger de uma ligação com inversão de sentido de fases.

Existe também o circuito de comando que será descrito no quinto capítulo.

## Capítulo 3

# PROCESSO DE CALANDRAGEM

O processo de conformação de chapa é composto por um conjunto de fases distintas, diferente do processo de corte, dobragem ou estampagem o processo de calandragem é semelhante ao processo de moldagem onde se o produto é construindo em várias fases. Como qualquer máquina a calandra também têm uma sequência lógica de trabalho, a dificuldade existente na conformação de chapa obrigou a que se desenvolve-se um processo de calandragem composto por varias fases ate ser atingido o formato pretendido na conformação da chapa.

Para se rentabilizar o fabrico de peças de formato cilíndrico, a necessidade de uma produção em série e garantia da qualidade do produto final, desenvolveu-se um modelo para o processo de calandragem dedicado à calandra de quatro rolos. Este modelo de calandragem é usado por qualquer operador de calandras com algum conhecimento de calandragem. A formalização do modelo de calandragem terá ocorrido na industria alemã, depois e universalizada nas grandes industrias metalomecânicas.

Com o aparecimento da máquina de quatro rolos de accionamento hidráulico, o processo de conformação ganha rapidez e perfeição no acabamento final.

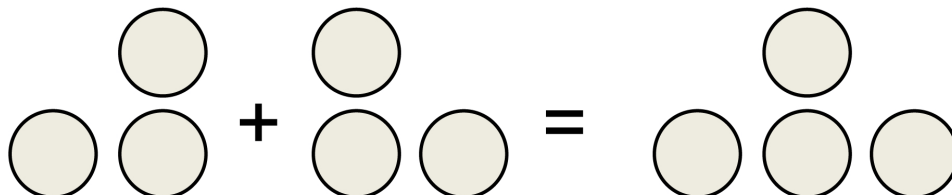


Figura 3.1: Topologia de três rolos e quatro rolos

Com a nova geometria é possível calandrar uma chapa metálica nas suas duas ex-

tremidades com uma só colocação, vantagem que não ocorre na calandra de três rolos assimétrica, outro factor de melhoramento é a secção recta nas extremidades da chapa que diminui significativamente em relação a uma maquina de três rolos piramidal.

O modelo de processo de calandragem é funcional, embora não seja em algumas fases bem quantificável em função de tipos de chapa ou diferentes mediadas, é necessário um ensaio para cada tipo de material ou mediadas pretendidas, podemos concluir que é bom para um trabalho em série onde o operador necessita de conhecimento prévio do comportamento do lote de chapa a calandrar.

Na implementação do modo automático de calandragem usando o modelo de processo de calandragem tradicional verifiquei grande dificuldade no controlo do processo, esta dificuldade ocorria na falta de experiência do sistema de comando com a variação das características de chapa a calandrar. Verifiquei que o processo de calandragem não era simétrico. Foi assim que surge a necessidade de contornar o sistema de calandragem tradicional de forma a que possamos aplicar um sistema automático capaz de produzir formas cilíndricas em chapa de varias dimensões sem necessidade de calibramos a máquina em função das características de cada chapa. Com a necessidade de criar um processo automático de calandragem mais eficiente e que depende-se menos das características da chapa foi desenvolvido um novo processo de calandragem onde foi obtido excelentes resultados.

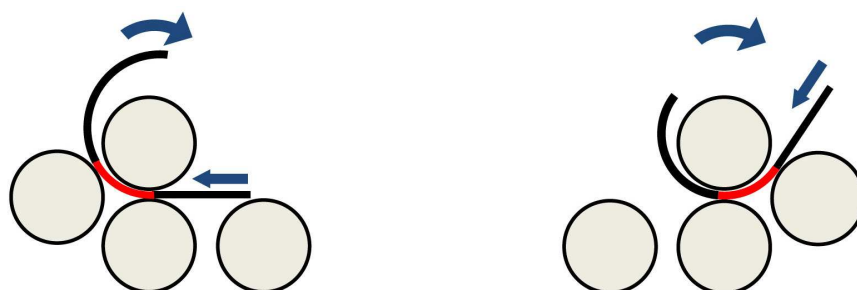


Figura 3.2: Dois Processos de Calandragem

## 3.1 Processo de Calandragem Tradicional

Na aplicação do método de calandragem tradicional podemos observar varias fases do processo de calandragem, podem ser divididas em oito fases que passamos a descrever como Figura 3.3:

- Fase 1 - Introduzir a Chapa Metálica

Esta é a primeira fase que dá início a uma sequência de calandragem, os rolos da máquina estão posicionados de forma a introduzir a chapa no seu interior.

- **Fase 2 - Apertar a Chapa entre o Rolo Superior e Inferior**

Nesta fase será importante verificar se a chapa ficou bem colocada, será dado o início da calandragem, o rolo inferior irá apertar a chapa contra o rolo superior.

- **Fase 3 - Rodar ao Início da Chapa**

Rodar a chapa para o início de forma a iniciar a calandragem de uma das extremidades.

- **Fase 4 - 1ª Parte de Calandragem**

A primeira parte de calandragem verifica-se aqui, subir o rolo direito e depois rodar para a esquerda, provoca assim a deformação na primeira parte da chapa.

- **Fase 5 - Passagem Intermedeia**

Aqui será a passagem de rolo, uma primeira parte calandrada com um rolo e uma segunda parte que irá ser calandrada com o rolo oposto.

- **Fase 6 - Início da Segunda Parte Calandragem**

Início da segunda fase de calandragem, rolo esquerdo em cima e rondar o restante da chapa.

- **Fase 7 - Calandragem 2ª Parte**

Nesta fase a calandra só roda para a esquerda.

- **Fase 8 - Fim de Calandragem**

A calandragem foi finalizada.

## 3.2 Processo de Calandragem Desenvolvido

Este processo é baseado numa necessidade de impor nas várias fases de calandragem uma simetria de posicionamento e rotação.

Analisando a calandra de quatro rolos, verificamos que esta é construída de forma simétrica, partindo desta característica e analisando as dificuldades de controlo no modelo tradicional, existe a necessidade de implementar um modelo de calandragem capaz de explorar a estrutura física da máquina, com o objectivo de contornar as dificuldades no

processo de calandragem, como o verificado no modelo tradicional, calandragem com o rolo direito ao rodar para a direita e com o rolo esquerdo ao rodar para a direita também ou o inverso. Devido a topologia da calandra quando rodamos para a direita e calandramos com o rolo da direita a deformação da chapa é feita depois da chapa passar os rolos principais (superior e inferior) mas se rodarmos para a esquerda e calandramos com o rolo da esquerda a deformação é provocada antes de passar os rolos principais.

Os factores apresentadas no paragrafo anterior apresentam algumas causas para as diferenças na conformação de uma chapa, não foi possível encontrar uma relação matemática para compensar o desvio provocado da primeira fase para a segunda fase, podemos observar a topologia na Figura 3.2. Esta sequência ocorre no processo de calandragem tradicional entre a fase quatro e a fase sete.

Para implementar um sistema automático de forma a contornar as falhas do modelo do processo de calandragem tradicional, parti de uma condição, sempre que seja feito um processo de deformação de chapa, ele ocorrerá depois de passar os rolos principais (superior e inferior), esta implementação está na base de um processo simétrico. Foi assim com a aplicação de um processo de calandragem simétrico para ambos os lados que criei um novo processo de calandragem. Este processo resulta num maior controlo sobre as varias fases do processo de calandragem, resultando num melhor controlo da calandra em modo automático.

As novas fases do processo de calandragem como mostra na Figura 3.4:

**- Fase 1 - Introduzir a Chapa Metálica**

Esta é a primeira fase que dá inicio a uma sequência de calandragem, os rolos da maquina estão posicionados de forma a introduzir a chapa no seu interior.

**- Fase 2 - Apertar a Chapa entre o Rolo Superior e Inferior**

Nesta fase será importante verificar se a chapa está bem colocada, será dado o inicio da calandragem, o rolo inferior irá apertar a chapa contra o rolo superior.

**- Fase 3 - Rodar a Chapa para o Meio**

Nesta fase será diferente do processo tradicional, aqui a chapa irá rodar no sentido directo 34 do diâmetro do rolo superior.

**- Fase 4 - Rodar Para Trás e Levantar Rolo Direito**

Fase em que se inicia a calandragem rodando no sentido inverso conformado com o rolo direito até atingir inicio da chapa.



- **Fase 5 - Primeira Parte Calandragem**

Fim da calandragem de um das extremidades da chapa.

- **Fase 6 - Rodar e Troca de Rolo de Calandragem**

Rodar ligeiramente no sentido directo para para inverter o sentido de calandragem.

- **Fase 7 - Calandragem da Segunda Parte Com Rolo Oposto**

Calandragem com posicionamento simétrico ao da fase quatro, irá até ao outro extremo da chapa para finalizar a calandragem.

- **Fase 6 - Fim de Calandragem**

Fim de calandragem.

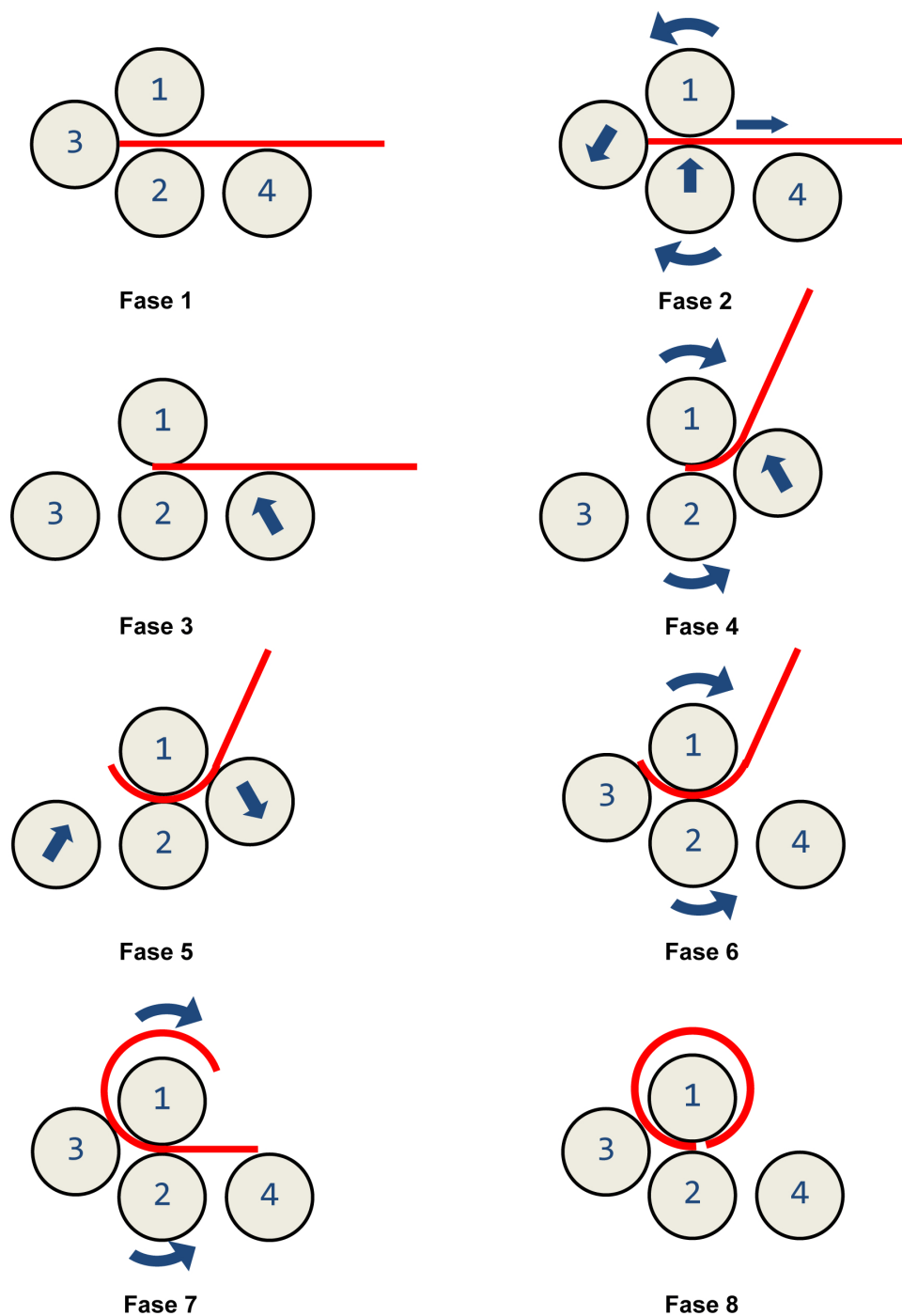


Figura 3.3: Fases de Calandragem Tradicional

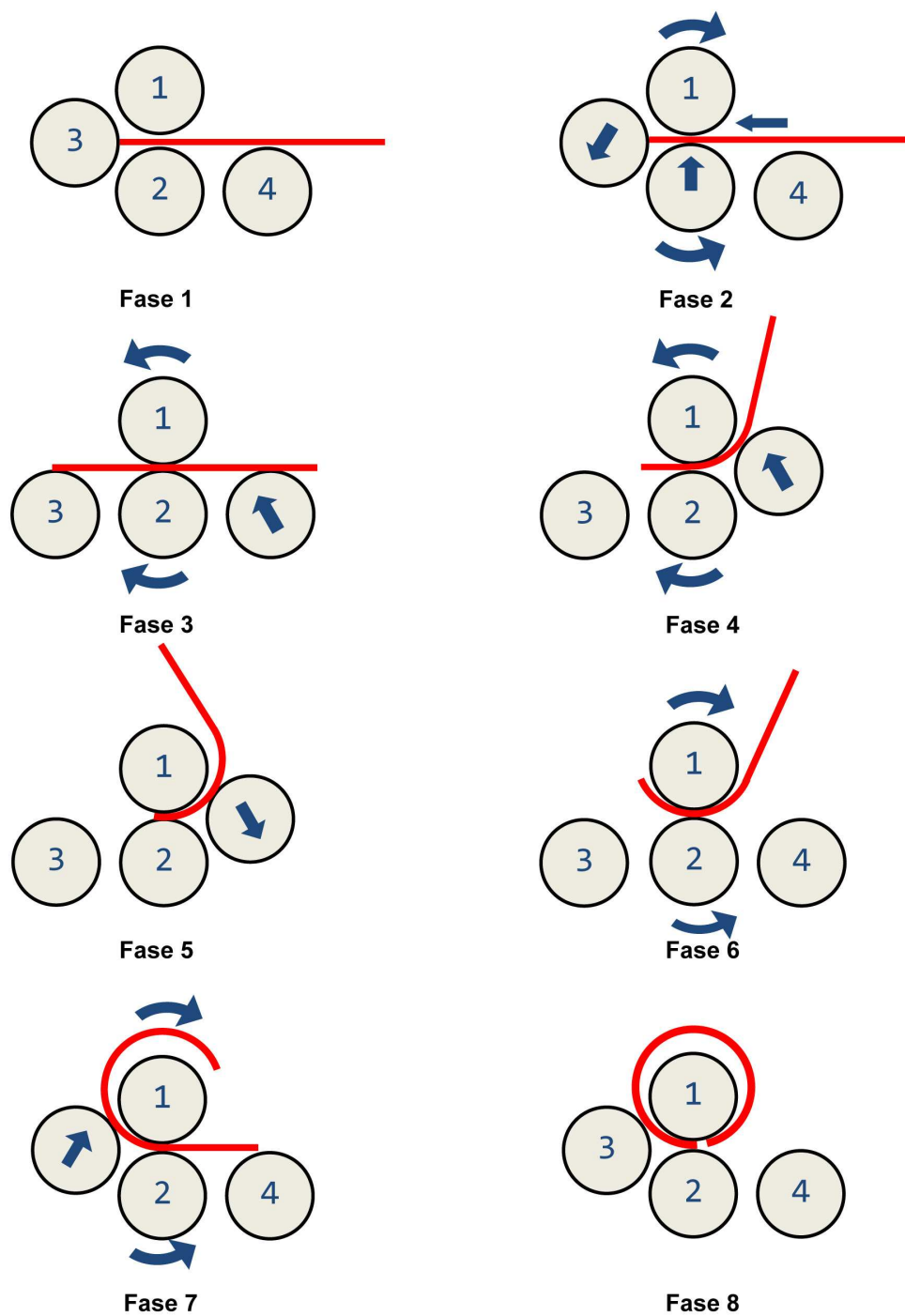


Figura 3.4: Fases de Calandragem do modelo Desenvolvido



## Capítulo 4

# ALGORITMO MATEMÁTICO DE POSICIONAMENTO

Neste capítulo vamos apresentar dois modelos matemáticos para ao cálculo do posicionamento dos rolos laterais no processo de conformação da chapa.

O primeiro algoritmo foi aplicado na primeira versão do sistema de controlo, desenvolvido no meu projecto de final de curso de licenciatura em Engenharia Electrónica e Telecomunicações do ano lectivo 2005/06, designado Controlo Automático de Calandras. O algoritmo foi construído baseado num modelo de calandra ideal, onde não estão contemplados factores como, desvios mecânicos ou de natureza da chapa, o algoritmo é a base de cálculo para o determinar a posição de um dos eixos(rolo lateral) em função do raio de curvatura pretendido. A experiência em calandragem adquirida nos últimos dois anos e o conhecimento adquirido da máquina levaram a que fosse possível contemplar outros factores intervenientes no processo de calandragem.

O segundo algoritmo de posicionamento foi desenvolvido com o objectivo de aproximar mais ao modelo de calandra real, embora se possa admitir que ainda não estão contemplados todos os parâmetros intervenientes na conformação da chapa, as alterações do primeiro algoritmo para o segundo produziram resultados práticos muito relevantes na conformação da chapa.

### 4.1 Primeiro Algoritmo de Posicionamento

Este algoritmo é baseado num modelo de calandra ideal em que não existem desvios mecânicos onde o aperto da chapa será de forma a que não existisse qualquer desvio da estrutura da máquina no posicionamento dos rolos.

O algoritmo matemático é baseado na Figura 4.1. Uma das grandes lacunas apresentadas por este algoritmo consistia na necessidade de ajustar a função de posicionamento em função de cada máquina.

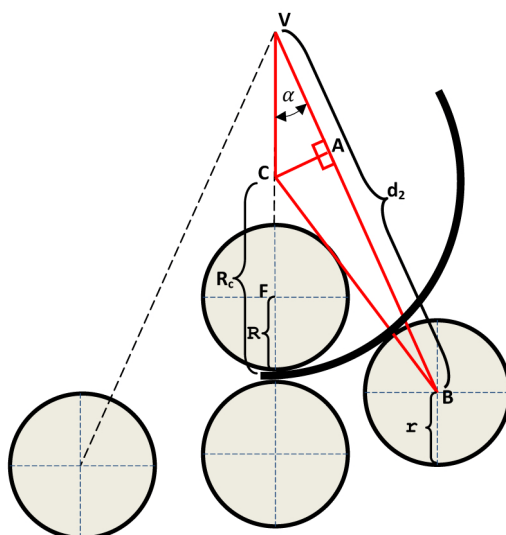


Figura 4.1: Diagrama da Calandra

Para calcular o posicionamento dos rolos laterais em função de um determinado raio de calandragem, o algoritmo baseia-se em pontos geométricos da estrutura da máquina que passo a descrever:

- **Vértice** ( $V$ ) - Ponto de intercepção do eixo do rolo inferior com os eixos dos rolos laterais.
- **Ângulo** ( $\alpha$ ) - Ângulo medido entre o eixo do rolo lateral esquerdo e o eixo do rolo inferior, que será de igual amplitude entre o eixo rolo lateral direito e o eixo do rolo inferior.
- **Raio Rolo Superior** ( $R$ ) - Raio do Rolo Superior, factor que é variável ao com a gama de calandras.
- **Raio Rolo Lateral** ( $r$ ) - Raio do Rolo Lateral, factor que é variável ao com a gama de calandras.
- **Distância do Vértice ao Centro do Rolo Superior** ( $VF$ ) - Distância entre o vértice e o centro do rolo superior, este valor é dado pelo projecto mecânico da máquina e é variável com o grupo das calandras.

- **Espessura** ( $e$ ) - Espessura da chapa que se está a programar.

O posicionamento do rolo lateral será em função da distancia ao ponto  $V$ .

**Formulação Matemática do Algoritmo:**

**Cálculos para o primeiro triângulo rectângulo:**

$$VC = (VF + R) - R_c$$

$$CA = VC \sin \alpha = ((VF + R) - R_c) \sin \alpha$$

$$VA = VC \cos \alpha = ((VF + R) - R_c) \cos \alpha$$

**Cálculos para o segundo triângulo rectângulo:**

$$CB = R_c + r + \frac{e}{2}$$

$$CA = VC \sin \alpha = ((VF + R) - R_c) \sin \alpha$$

**Aplicação do teorema de Pitágoras:**

$$CB^2 = CA^2 + AB^2 \implies AB^2 = CB^2 - CA^2$$

$$AB = \sqrt{CB^2 - CA^2} = \sqrt{(R_c + r + \frac{e}{2})^2 - (((VF + R) - R_c) \sin \alpha)^2}$$

**Expressão final para o posicionamento em função do Raio  $R_c$  :**

$$d_2 = VA + AB$$

Temos a equação final do posicionamento do rolo lateral em função do raio de curvatura pretendido e dos restantes parâmetros da geometria da máquina.

$$d_2 = ((VF + R) - R_c) \cos \alpha + \sqrt{(R_c + r + \frac{e}{2})^2 - (((VF + R) - R_c) \sin \alpha)^2} \quad (4.1)$$

A distância  $d_2$  está em função do ponto  $V$ , e será a distância entre  $V$  e o ponto  $B$ . Por motivos práticos foi definido a origem dos vários eixos de posição em relação ao ponto máximo de elevação, foi implementado um sistema de zero relativo nos pontos de intercepção de cada rolo com o superior, como podemos observar na Figura 4.2. Em resultado desta definição de origem dos eixos, é mais prático que o algoritmo de posicionamento

esteja em função da origem definida para cada eixo. Para isso basta calcular o posicionamento dos rolos na sua origem em função do ponto  $V$  e relacionar com a Equação 4.1. O posicionamento dos rolos laterais no ponto de origem é considerar o valor de  $R_c = R$  e o valor da espessura  $e = 0$ , valores usados para o calculo de posicionamento na origem.

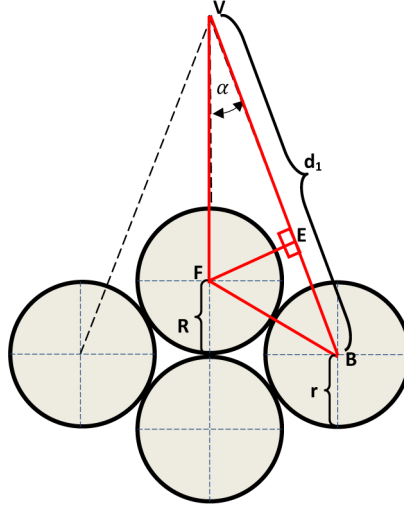


Figura 4.2: Rolos na Posição de Origem Referência Zero dos Eixos

#### Formulação Matemática da Origem dos Eixos dos Rolos Laterais:

$$d_1 = ((VF) \cos \alpha + \sqrt{(R + r)^2 - ((VF) \sin \alpha)^2}) \quad (4.2)$$

Temos a equação final de posicionamento obtida pela diferença de  $d_2$  por  $d_1$ , o valor obtido será sempre entre o valor zero e o maior diâmetro possível que será um valor que não irá provocar deformação da chapa, nesta posição o rolo estará paralelo ao rolo inferior.

#### Formula Matemática da Posição do Rolo Lateral em Função da Origem:

$$d = d_2 - d_1$$

$$d = ((VF + R) - R_c) \cos \alpha + \sqrt{(R_c + r + \frac{e}{2})^2 - (((VF + R) - R_c) \sin \alpha)^2} - ((VF) \cos \alpha + \sqrt{(R + r)^2 - ((VF) \sin \alpha)^2}) \quad (4.3)$$



## 4.2 Segundo Algoritmo de Posicionamento

No segundo algoritmo é introduzido um desvio angular do eixo referente ao rolo inferior, onde se encontra o centro da circunferência programada o ponto  $C$  da Figura 4.1, este desvio é devido à geometria dos rolos da calandra e provocado pelos pontos de apoio da chapa serem muito pequenos, a chapa é apertada entre dois rolos cilíndricos onde tende a existir escorregamento dos pontos de contacto, podemos observar o efeito na Figura 4.3. O efeito verificado poderá ser minimizado se a pressão de aperto da chapa for mais elevada, mas devido à superfície de contacto ser muito pequena essa força de aperto irá provocar deformação na chapa, por este motivo o posicionamento não pode ultrapassar a medida da espessura da chapa. O fenómeno verificado também é devido à elasticidade da chapa que numa primeira fase da conformação, não irá existir ruptura das suas ligas internas enquanto não se ultrapassa a resistência elástica do material. As calandras produzidas pela empresa Marcovil, S.A. são dimensionadas para uma chapa do tipo St 37 contemplada na norma DIN ST 37-2 que possui características mecânicas de:  $ReMpa = 235$  e  $RmMpa$  dentro de um intervalo  $[350 - 480]$ . O desvio angular provocado da Figura 4.1 para a Figura 4.4 é também em função da carga necessária para vencer a resistência elástica da chapa, como referi no paragrafo anterior, que neste tipo de chapa será mínimo de  $235N/mm^2$ .

Não são só os factores que falamos que intervêm na conformação da chapa, mas toda a estrutura mecânica da maquina composta por folgas e deslizes naturais deste tipo de maquinas, bem como a irregularidade existente nas superfícies das chapas e temperatura. A chapa normalmente usada nas metalomecânica é chapa laminada a quente que não apresenta uniformidade de natureza nem de demissões, embora se recomende o uso de chapa certificada para garantir um menor desvio em relação à chapa ideal, o que se verifica é que a relação custo/tempo compensa para a maioria dos trabalhos o uso de chapa não certificada.

O desvio angular referenciado é determinado para cada gama maquina, baseando-se em testes práticos suportados nos parâmetros mecânicos na chapa, e com este processo se faz uma calibração da maquina em função do meio que está a ser utilizada. Normalmente a maquina é calibrada na fase de testes e mais tarde na instalação da maquina no cliente, onde é calibrada em função do tipo de material utilizado pelo cliente.

Em resultado do observado na Figura 4.3 e em conjugação com o primeiro algoritmo obtive-se o algoritmo apresentado na Figura 4.4

O posicionamento do rolo lateral será em função da distância ao ponto  $G$  da Figura 4.4.

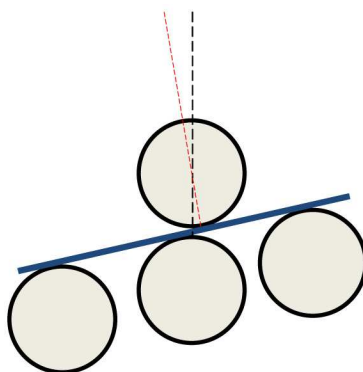


Figura 4.3: Efeito de Deslocação dos Pontos de Contacto

### Formulação Matemática do Algoritmo:

**Cálculos para o primeiro triângulo rectângulo:**

$$HV = (VF + R + \frac{\epsilon}{2}) \sin \beta$$

$$\gamma = \alpha - \beta$$

$$GV = \frac{HV}{\sin \gamma}$$

$$GH = \frac{HV}{\cos \gamma} = \frac{(VF + R + \frac{\epsilon}{2}) \sin \beta}{\cos \gamma}$$

**Calculos para o segundo triângulo rectângulo:**

$$HI = (VF + R + \frac{\epsilon}{2}) \cos \beta$$

$$GI = HI + GH = (VF + R + \frac{\epsilon}{2}) \cos \beta + \frac{(VF + R + \frac{\epsilon}{2}) \sin \beta}{\cos \gamma}$$

$$GC = GI - R_c = ((VF + R + \frac{\epsilon}{2}) \cos \beta + \frac{(VF + R + \frac{\epsilon}{2}) \sin \beta}{\cos \gamma}) - R_c$$

$$GA = GC \cos \gamma = (((VF + R + \frac{\epsilon}{2}) \cos \beta + \frac{(VF + R + \frac{\epsilon}{2}) \sin \beta}{\cos \gamma}) - R_c) \cos \gamma$$

$$CA = GC \sin \gamma = (((VF + R + \frac{\epsilon}{2}) \cos \beta + \frac{(VF + R + \frac{\epsilon}{2}) \sin \beta}{\cos \gamma}) - R_c) \sin \gamma$$

$$CB = R_c + r + \frac{\epsilon}{2}$$

**Aplicação do teorema de Pitágoras:**

$$CB^2 = AB^2 + CA^2 \implies AB^2 = CB^2 - CA^2$$

$$AB = \sqrt{CB^2 - CA^2}$$

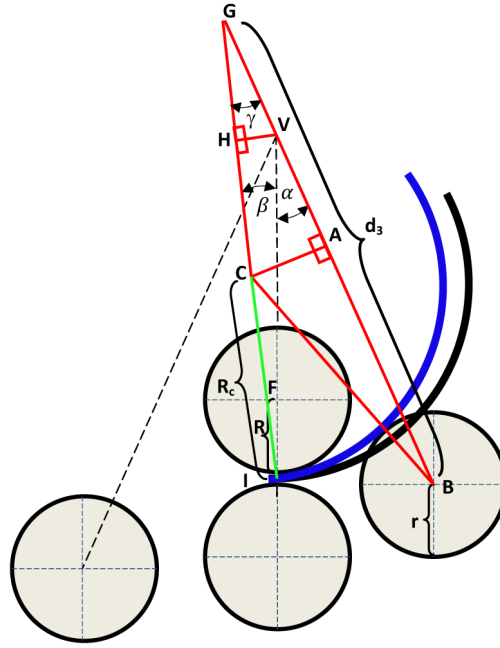


Figura 4.4: Diagrama da Calandra Modificado

$$AB = \sqrt{(R_c + r + \frac{e}{2})^2 - (((VF + R + \frac{e}{2}) \cos \beta + \frac{(VF + R + \frac{e}{2}) \sin \beta}{\cos \gamma}) - R_c) \sin \gamma)^2}$$

**Expressão final para o posicionamento em função do Raio pretendido**

$R_c$  :

$$d_3 = GA + AB$$

Temos a equação final do posicionamento do rolo lateral em função do raio de curvatura pretendido e restantes parâmetros da geometria da máquina.

$$d_3 = (((VF + R + \frac{e}{2}) \cos \beta + \frac{(VF + R + \frac{e}{2}) \sin \beta}{\cos \gamma}) - R_c) \cos \gamma + \sqrt{(R_c + r + \frac{e}{2})^2 - (((VF + R + \frac{e}{2}) \cos \beta + \frac{(VF + R + \frac{e}{2}) \sin \beta}{\cos \gamma}) - R_c) \sin \gamma)^2} \quad (4.4)$$

Para que a expressão de posicionamento venha em função das origens dos eixos, vamos calcular a posição para colocar os rolos na posição de origem e subtrair á Equação 4.4 para assim obtermos a expressão final em função do valor de origem de cada eixo.

Expressão na origem será colocar o parâmetro  $R_c = R$  e  $e = 0$ , obten-se:

$$d_3 = (((VF + R) \cos \beta + \frac{(VF + R) \sin \beta}{\cos \gamma}) - R) \cos \gamma + \sqrt{(R + r)^2 - (((VF + R) \cos \beta + \frac{(VF + R) \sin \beta}{\cos \gamma}) - R) \sin \gamma)^2} \quad (4.5)$$

Expressão final que resulta da subtracção da Equação 4.5 na Equação 4.4.

$$\begin{aligned} d = & (((VF + R + \frac{e}{2}) \cos \beta + \frac{(VF + R + \frac{e}{2}) \sin \beta}{\cos \gamma}) - R_c) \cos \gamma + \\ & + \sqrt{(R_c + r + \frac{e}{2})^2 - (((VF + R + \frac{e}{2}) \cos \beta + \frac{(VF + R + \frac{e}{2}) \sin \beta}{\cos \gamma}) - R_c) \sin \gamma)^2 -} \\ & - (((VF + R) \cos \beta + \frac{(VF + R) \sin \beta}{\cos \gamma}) - R) \cos \gamma + \\ & + \sqrt{(R + r)^2 - (((VF + R) \cos \beta + \frac{(VF + R) \sin \beta}{\cos \gamma}) - R) \sin \gamma)^2} \quad (4.6) \end{aligned}$$

## Capítulo 5

# AUTOMATIZAÇÃO DA CALANDRA HIDRÁULICA

Com a necessidade de implementar um sistema automático capaz de controlar todo o processo posicionamento dos cilindros e rotação dos rolos em função de um raio de curvatura, o controlo da rotação é o factor fundamental necessário para permitir a evolução do sistema de controlo da calandra.

Nas máquinas de 3 rolos não era possível controlar a progressão da chapa, pelo facto de esta não permanecer em contacto permanente entre dois rolos, no processo de calandragem da máquina de três rolos existem momentos de escorregamento e momentos em que a chapa está em contacto permanente entre dois rolos. Para a máquina de 4 rolos o sistema de contacto directo com a chapa é diferente, devido ao facto de existir dois rolos com a mesma cota de eixo dos  $xx$  que permite um contacto permanente dos dois rolos a chapas, a distância entre o rolo inferior e o superior é normalmente a espessura da chapa.

Colocando um dispositivo de contagem acoplado ao rolo superior, será possível controlar o eixo da rotação.

Na automatização e implementação de sistemas de controlo numérico é bastante vulgar o termo de eixo de controlo, para o desenvolvimento de este trabalho foi definido vários eixos de controlo que passaram a ser apresentados na secção seguinte. Um eixo de controlo é uma direcção de referencia que definimos para um determinado deslocamento, será um sistema de coordenadas para todos os movimentos necessários de controlar numericamente pelo sistema de comando. No caso da calandra vamos definir como eixos principais os deslocamentos de todos os cilindros hidráulicos e a rotação dos rolos da máquina.

## 5.1 Implementação dos Eixos de Controlo

Um dos primeiros passos necessários para a automatização é definir os eixos intervenientes no sistema a controlar. Para o processo de conformação da chapa é necessário termos controlo sobre o posicionamento dos rolos laterais, porque será o posicionamento dos rolos laterais que provocam a deformação da chapa, o posicionamento do rolo inferior que irá garantir a distância entre o rolo superior e inferior, a rotação será o eixo que falta para que o processo se possa realizar num modo automático.

A calandra está construída para conformar chapa em formas cilíndricas ou conicas. Para que se possa realizar na calandra peças com formato cónico, a calandra têm que permitir que um dos extremos dos rolos seja accionado separadamente, por esta necessidade implica que cada rolo tenha que possuir dois eixos. Como cada rolo está suportado nas suas duas extremidades por cilindros hidráulico, podemos quantificar 2 eixos, um eixo por cilindro hidráulico, assim quando necessitamos de subir ou descer, podemos realizar essa tarefa com base numa distância, o mesmo processo é feito para os restantes 2 rolos, no total são definidos 6 eixos de posição o que nos permite implementar controlo de posicionamento.

Para podermos controlar o processo de calandragem é necessário ter um eixo de rotação, como já vimos a traz é possível implementar um eixo de rotação nas calandras de 4 rolos devido a sua geometria, assim com a colocação de um encoder de rotação acoplado ao rolo superior temos o eixo de rotação.

Uma Calandra de 4 rolos base é composta por 7 Eixos, um de rotação e 6 de posição, sendo 2 por rolo inferior, 2 por o rolo lateral direito e 2 pelo rolo lateral esquerdo. Na totalidade é necessário contagem para 7 eixos, sendo cada eixo composto por dois sinais *A* e *B* com uma forma de onda quadrada, os sinais são lidos em quadratura para que se possa diminuir o erro de leitura.

### 5.1.1 Sinais de Eixos

Existe a necessidade de identificar o sistema de eixos para uma boa implementação do sistema de controlo.

Tende em vista o controlo em modo automático foi necessário determinar todos os eixos relevantes para o processo de calandragem. Para atingirmos os objectivos foi necessário observar a calandra a trabalhar em modo manual, deste modo verificamos que é necessário sete eixos para uma calandra base e dez eixos para uma calandra completa equipada com suportes laterais e torre de apoio vertical.

Temos então o nosso sistema de eixos compostos por:

- (1E) Eixo de posicionamento Rolo Esquerdo (Trás)
- (2E) Eixo de posicionamento Rolo Esquerdo (Frente)
- (3D) Eixo de posicionamento Rolo Direito (Trás)
- (4D) Eixo de posicionamento Rolo Direito (Frente)
- (5I) Eixo de posicionamento Rolo Inferior (Trás)
- (6I) Eixo de posicionamento Rolo Inferior (Frente)
- (RT) Eixo de posicionamento Rotação
- (SV) Eixo de posicionamento Suporte Vertical
- (SE) Eixo de posicionamento Suporte Lateral Esquerdo
- (SD) Eixo de posicionamento Suporte Lateral Direito

## 5.2 Definição de Sinais *Input/Output*

Outra das fases fundamentais para implementarmos qualquer sistema de controlo, é determinar os sinais de *input*(entrada) e *output*(saída).

No caso da Calandra vamos fazer a análise para a calandra de 4 rolos visto ser a mais completa e a que é possível adicionar o sistema de controlo mais completo.

### 5.2.1 Sinais de *Input*

Os sinais de entrada são maioritariamente oriundos do comando de accionamento manual, a calandra está preparada para ser accionada manualmente por joysticks que permitem subir e baixar os rolos laterais e o rolo inferior, outros comandos importantes são os que permitem accionar a rotação dos rolos para no sentido directo ou inverso, existem outros que passamos a descrever:

- (01) Subir Rolo Lateral Esquerdo
- (02) Subir Rolo Lateral Direito
- (03) Subir Rolo Inferior
- (04) Baixar Rolo Lateral Esquerdo

- (05) Baixar Rolo Lateral Direito
- (06) Baixar Rolo Inferior
- (07) Rodar Sentido Directo
- (08) Rodar Sentido Inverso
- (09) Fechar Braço de Apoio
- (10) Abrir Braço de Apoio
- (11) Subir Suporte Vertical
- (12) Baixar Suporte Vertical
- (13) Subir Suporte Lateral Esquerdo
- (14) Baixar Suporte Lateral Esquerdo
- (15) Subir Suporte Lateral Direito
- (16) Baixar Suporte Lateral Direito
- (17) Activar Modo Conico
- (18) Botão de Start
- (19) Botão de Stop
- (20) Botão ON/OFF
- (21) Sinal de Fim de Curso Suporte Rolo Superior
- (22) Sinal de Falha de Tensão de Alimentação
- (23) Sinal de Rotação Lenta
- (24) Sinal de Posicionamento Lento
- (25) Sinal de Nivel Hidráulico

Todos estes sinais podem ser classificados em seis grupos que passamos a classificar de sinais principais de posição, sinais principais de rotação, sinais secundários de posição, sinais de segurança e sinais de controlo.



### 1. Sinais Principais Posição

- Subir Rolo Lateral Esquerdo
- Subir Rolo Lateral Direito
- Subir Rolo Inferior
- Baixar Rolo Lateral Esquerdo
- Baixar Rolo Lateral Direito
- Baixar Rolo Inferior

Estes são os sinais que permitem accionar o posicionamento manual dos rolos laterais e do rolo inferior tanto na subida como na descida.

### 2. Sinais Principais Rotação

- Rodar Sentido Directo
- Rodar Sentido Inverso

Estes são os sinais que permitem accionar a rotação manual de todos os rolos para a direita e para a esquerda.

### 3. Sinais Secundário Posição

- Fechar Braço de Apoio
- Abrir Braço de Apoio
- Subir Suporte Vertical
- Baixar Suporte Vertical
- Subir Suporte Lateral Esquerdo
- Baixar Suporte Lateral Esquerdo
- Subir Suporte Lateral Direito
- Baixar Suporte Lateral Direito

Estes sinais permitem abrir e fechar o suporte do rolo superior e também para baixar e subir os suportes laterais e superiores, todos estes comandos são accionados manualmente.

### 4. Sinais de Segurança

- Sinal de Fim de Curso Braço de Apoio
- Sinal de Falha de Tensão de Alimentação
- Sinal de Nivel Hidráulico

Estes são uns dos sinais principais para o funcionamento da calandra, o primeiro é referente á abertura do suporte do rolo superior, caso este suporte esteja aberto os comandos principais da calandra não funcionam, como o posicionamento e a rotação. Outro sinal é a detecção de falha de energia, neste caso o sistema de comando irá suspender-se, este processo é feito com o suporte da energia da de backup da unidade de armazenamento de energia.

## 5. Sinais de Controlo

- Activar Modo Conico
- Botão de Start
- Botão de Stop
- Botão ON/OFF
- Sinal de Rotação Lenta
- Sinal de Posicionamento Lento

Os sinais de controlo são usados para definir métodos de trabalho, como trabalhar em modo cónico, neste modo o posicionamento dos rolos é feito do lado de topo da maquina o que permite um desnível nos rolos, podendo assim fazer-se peças de formato cónico, os outros sinais permitem fazer pausas e stops e iniciar processos automáticos, e também permite dar ordem para que a calandra nivele automaticamente os rolos, por fim os últimos sinais permitem seleccionar velocidades de posicionamento e rotação diferentes.

O numero normal por maquina é de vinte e cinco sinais de entrada.

### 5.2.2 Sinais de Output

Os sinais de output são maioritariamente sinais que irão actuar as electroválvulas direccionais, os sinais de saída foram definidos em 24 VDC que alimentam bobinas de reles PLC's.

Os reles PLC servem de interface entre o accionamento das electroválvulas e o sistema de comando, isto é devido a dois factores fundamentais para o bom funcionamento do sistema:

1. A colocação dos reles PLC's permite isolar o sistema de comando do resto da estrutura e assim proteger o hardware de comando que normalmente é mais sensível.
2. Devido á instalação da maquina ser elevada e á necessidade de uma elevada potência para activar as electroválvulas, a tensão de alimentação das electroválvulas é de 24 VAC, pelo motivo que é mais económico um transformador de 700 VA que uma fonte de 24 VDC para a mesma potência, assim os reles estão também com a função de actuadores de DC para AC.

Como os sinais de entrada também os sinais de saída se podem classificar por grupos assim temos:

### **1. Sinais Electroválvulas Principais Posicionamento**

- (S19) Electroválvula S19
- (S20) Electroválvula S20

Estas duas electroválvulas são as principais do posicionamento, ou seja são as duas electroválvulas que recebem o fluxo hidráulico das duas bombas principais e alimentam separadamente as duas extremidades da Calandra, uma para todos os macacos da parte da frente e outra para os macacos do lado dos motores.

### **2. Sinais Electroválvulas Principais Posicionamento Lento**

- (19L) Electroválvula 19L
- (20L) Electroválvula 20L

Estes sinais são para seleccionar o fluxo hidráulico proveniente de duas bombas de caudal reduzido que permite fazer as aproximações aos pontos de paragem com velocidades reduzidas, também como nos sinais anteriores este fluxo hidráulico irá alimentar as duas extremidades da Calandra.

### **3. Sinais Electroválvulas Posição Lado dos Motores**

- (S03) Subir Rolo Esquerdo Trás
- (S04) Descer Rolo Esquerdo Trás
- (S07) Subir Rolo Inferior Trás
- (S08) Descer Rolo Inferior Trás

- (S05) Subir Rolo Direito Trás
- (S06) Descer Rolo Direito Trás

Estes são os sinais que permitem subir ou baixar todos os macacos do lado de traz ou seja do lado dos motores da maquina.

#### 4. Sinais Electroválvulas Posição Lado da Frente

- (S09) Subir Rolo Esquerdo Frente
- (S10) Descer Rolo Esquerdo Frente
- (S13) Subir Rolo Inferior Frente
- (S14) Descer Rolo Inferior Frente
- (S11) Subir Rolo Direito Frente
- (S12) Descer Rolo Direito Frente

Estes são os sinais que permitem subir ou baixar todos os macacos do lado de frente ou seja do lado da abertura da maquina.

#### 5. Sinais Electroválvulas Rotação

- (S01) Rolo Superior Direita
- (S02) Rolo Inferior Direita
- (S01A) Rolos Laterais Direita
- (S02A) Rolo Superior Esquerda
- (S01B) Rolo Inferior Esquerda
- (S02B) Rolos Laterais Esquerda

Estes são os sinais que permitem fazer rodar os rolos no sentido directo e inverso.

#### 6. Sinais Electroválvulas Posição Suportes

- (S30) Subir Suporte Vertical
- (S31) Baixar Suporte Vertical
- (S32) Subir Suporte Lateral Esquerdo
- (S33) Baixar Suporte Lateral Esquerdo
- (S34) Subir Suporte Lateral Direito

- (S35) Baixar Suporte Lateral Direito

Estes são os sinais que permitem fazer subir e baixar os suportes laterais e verticais.

#### 7. Sinais Electroválvulas Braço de Apoio

- (S15) Abrir Suporte Rolo Superior
- (S16) Fechar Suporte Rolo Superior

Estes são os sinais que permitem fazer abrir e fechar o suporte do rolo superior.

#### 8. Sinais de Controlo Velocidade

- Variação de Rotação

Este sinal permite a comutação de velocidades .

#### 9. Sinais de Controlo Energia e PC Industrial

- Activar Unidade de Energia
- Activar PC Industrial

Estes sinais permitem a comutação da unidade de backup de energia e ligação do PC Industrial.

O Numero de sinais de saída típicos por calandra são trinta e um sinais maioritariamente de 24 VDC que conectam em reles PLC's, onde é feito um isolamento de saída dos hardware de comando pelos próprios reles. O sinal comum dos contactos dos reles PLC estão ligados a 24 VAC que por sua vez ligam em cachimbos rectificadores das electroválvulas, deste modo é feito o sistema de protecção do comando, isolando o sistema com reles.

### 5.3 Estrutura *Hardware* Existente

No decorrer do desenvolvimento do sistema de controlo foram avaliadas várias soluções de *hardware*. Quando foi proposto pela primeira vez na Universidade de Aveiro no âmbito da cadeira de projecto de final de curso, desenvolver um sistema de comando a implementar numa calandra hidráulica no ano de 2005/06, a empresa Marcovil,SA na qual eu estou a realizar este trabalho já tinha implementado uma primeira estrutura de comando. No decorrer do projecto de final de curso de Licenciatura em Eng. Electrónica e Telecomunicações no ano lectivo 2005/06 eu e o meu colega iniciamos o desenvolvimento de uma

nova estrutura com o objectivo de possibilitar suporte para implementar um comando do tipo CNC (*computer numerical control*). Neste projecto foi dimensionado um primeiro sistema de *hardware* onde verificamos alguma dificuldade em controlar o sistema, embora não tenha resultado em pleno com o nosso trabalho foi construída a base de trabalho para os futuros dimensionamentos do sistema de comando.

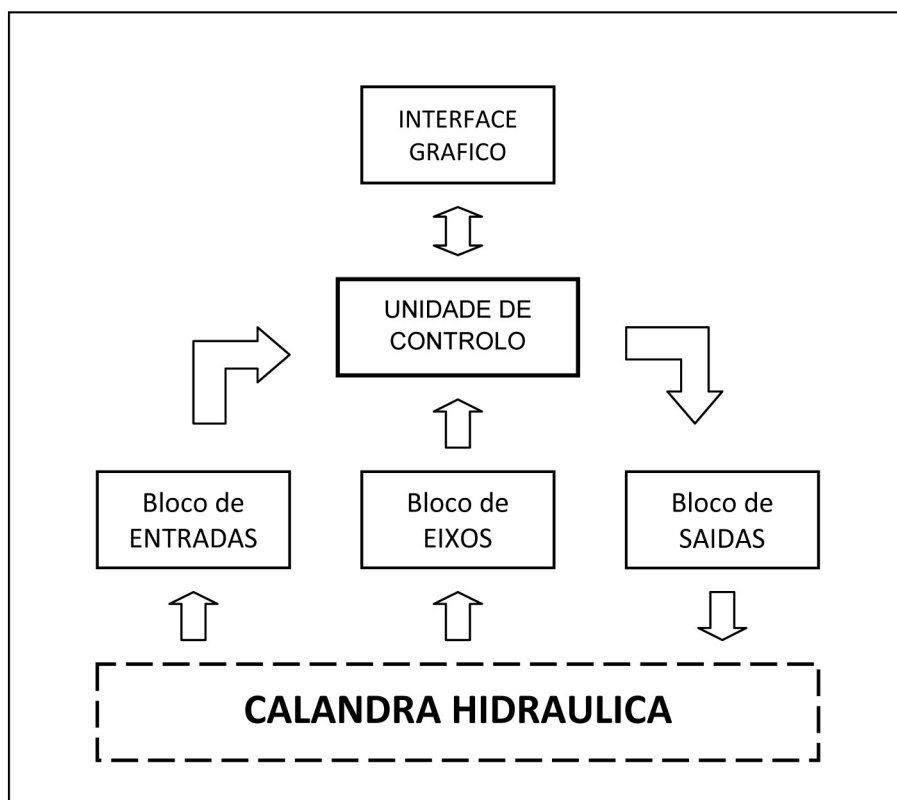


Figura 5.1: Diagrama Blocos Sistema Controlo

### 5.3.1 *Hardware* do 1º Sistema de Comando

Como referi atrás existiu um primeiro sistema de comando aplicado á calandra. O primeiro sistema de comando criado foi desenvolvido na empresa Marcovil,SA onde eu hoje continuo desenvolvendo o actual sistema de controlo. Nesta primeira solução foi implementado uma base de *hardware* actual na sua época e que foi importante para chegarmos aos dias de hoje e apresentar soluções com muito mais capacidades.

#### Composição do 1º Sistema de Comando

- Monitor TFT 12.1"



Figura 5.2: Sistema Controlo Antigo

- Teclado Industrial
- Processador 485 MMX
- 3 Placa Digitais ISA de 8 Entradas
- 3 Placa Digitais ISA de 8 Saídas
- 3 Placa Digitais ISA de 4 Eixos

Este sistema era muito limitado, o seu *hardware* era uma estrutura composta por um computador com poucos recursos para controlar todo o sistema. O Sistema de controlo era baseado no software *MS Dos 6.22* e com uma aplicação desenvolvida em *Turbo C* que geria o sistema de controlo. Foi desenvolvido um interface homem-maquina baseado nos suportes gráficos do *Ms Dos*, todo este sistema era funcional para a época que foi implementado. Uma das grandes lacunas neste sistema foi o pouco desenvolvimento verificado na interpolação de eixos para permitir um controlo do processo de conformação de chapa. O sistema de comando em modo automático era pouco eficiente e na grande maioria não se obtinha resultados satisfatórios no processo de produção.

O algoritmo de posicionamento em função de um determinado valor de raio era muito pouco eficiente o que levava a que os operadores deste tipo de maquinas trabalha-sem em modo manual e não em modo automático.

Outro factor que não era favorável ao sistema de comando é o seu custo que estava avaliado em 8000.00*Euros* por unidade.

O sistema era composto por uma *matherboard* com um barramento ISA onde eram conectados a placa gráfica e as placas de entrada e saída e as placas de contagem, na totalidade eram 6 placas conectadas ao barramento ISA, dispunha de um monitor e de uma unidade de disco com pouca capacidade de armazenamento. Por falta de espaço para armazenamento não era possível armazenar programas na máquina, outra inconveniência que ocorria era a falha de contacto que surgia no barramento ISA com as placas, esta falha de contacto era provocada pelo movimento do painel de comando móvel muito usual para este tipo de máquina.

Com todas as limitações que se encontra neste sistema a Marcovil,S.A. fez o pedido de na Universidade de Aveiro para desenvolver um sistema de novo comando com a finalidade de implementação nas futuras calandras hidráulicas.

### 5.3.2 *Hardware* do Sistema de Comando Desenvolvido em 2005/06

Nesta fase que foi desenvolvida no âmbito do projecto de final de curso de Licenciatura de Eng. Electrónica e Telecomunicações, foi dimensionado uma estrutura de *hardware* para suportar um *software* mais poderoso, que permiti-se um pleno controlo sobre o processo de conformação de chapa e ao mesmo tempo suporta-se aplicações do tipo *API Application Programming Interface*, outro requisito para uma boa aceitação perante os utilizadores é a implementação de um sistema *user frendly* de modo a melhor argumentos de mercado.

Perante estas exigências foi definida uma estrutura de hardware centralizado baseado num computador actual com placas de entrada e saída em barramento PCI, para o sistema operativo foi definido o MS Windows XP Embedded por ser mais pequeno e ter as potencialidades necessárias e também por ser necessário instalar todo o *software* num cartão de memória, esta necessidade é uma medida de segurança visto que este comando será aplicado a uma máquina industrial, onde será necessário que não existam partes moveis no sistema de comando para diminuir a probabilidade de avarias. Um sistema operativo funcional foi instalado num cartão de memória de 512MB.

#### Composição do 2º Sistema de Comando

- Monitor TFT 12.1"Touch Screen
- Teclado Industrial
- Processador P4 2.4GHZ



- 2 Placa Digitais PCI de 16 Entradas/ 16 Saídas/8 EIXOS

Esta solução tinha como objectivos a redução de custos e o incremento de qualidade e recursos para suportar um novo *software* capaz de se impor como um comando CNC. Uma das limitações deste sistema foi o sistema operativo Ms Windows Xp Embedded, a estrutura de contagem dos eixos foi definida para que essa contagem se realiza-se por *software* como o sistema operativo não é de tempo real não permite agendar rotinas de contagem com tempos menores que 10ms e garantir que existe uma politica de gestão de tempos de processos e mesmo permitir uma politica de prioridades para evitar *deadlock* ou adiamentos indefinidos. Quando definimos a contagem dos eixos por software deveríamos ter definido o sistemas de contagem por *hardware* todo o processo seria mais simples de implementar, deveríamos ter em conta que esta máquina possui dez eixos embora a frequência máxima não chegue aos 5kHz por eixo.

Esta solução desenvolvida não chegou a ser uma solução final para implementação numa calandra por motivos de falta de tempo para otimizar todo o que se tinha desenvolvido.

Foram detectados algumas deficiências mas na globalidade o projecto foi apresentando soluções para solucionar as questões que ocorriam na prática, com o terminar do projecto no âmbito da cadeira de projecto foi iniciada uma segunda fase do projecto mas esta já nas instalações da empresa Marcovil,S.A.

Nesta segunda fase tinha como objectivos chegar a solução final no sistema de comando, para se iniciar a instalação nas novas maquinas que se estavam a produzir. Foi assim que eu passei a colaborador da empresa Marcovil,S.A. continuando o desenvolvimento do sistema de comando iniciado no projecto de final de curso.

## 5.4 Estrutura de *Hardware* do Sistema de Comando Distribuindo

No desenvolvimento da solução final foi necessário definir uma estrutura de *hardware* que permita o suporte de todas as aplicações idealizadas para o presente e que possa suportar a evolução do sistema mantendo a estrutura base. Existiu a necessidade de desenvolvermos dois tipos de comando, um mais avançado outro uma gama mais económica que permita alguma vantagem perante um painel de comando simples.<sup>1</sup>

Por motivos de existência de dois tipos de comando foi definido uma estrutura de comando global de onde deriva os dois comandos. O sistema de comando mais completo

---

<sup>1</sup>Comando que só possui botões e manípulos para accionar o sistema hidráulico da maquina

foi projectado para ser do tipo CNC<sup>2</sup> e por ser o mais importante será o que irá ter mais peso na definição da estrutura de *hardware*, o outro sistema de comando desenvolvido foi do tipo NC<sup>3</sup> que será composto por uma estrutura de hardware mais simples.

Um dos objectivos é permitir que exista uma só conexão à calandra hidráulica, e a esta conexão pode-se ser ligado um ou outro comando e ambos executassem as suas funções em pleno. De uma maneira simples de distinguir os dois sistemas projectados será pela ausência do computador industrial no comando NC e em consequência a diminuição de capacidade em relação ao comando CNC.

### 5.4.1 Estrutura de *Hardware* do Sistema de Comando CNC

Pela complexidade ao nível do número de eixos existentes neste tipo de máquina e projectando a máquina para que possa existir interpolação entre eixos na execução da conformação de chapa. Foi implementado uma estrutura de comando distribuído. Outros argumentos nos levaram a implementarmos esta estrutura como: tornar o bloco central mais livre, simplificar uma identificação de falhas, possibilitar adição de módulos sem termos projectar novo sistema, permitir funcionamentos básicos da máquina quando ocorre uma falha no sistema de controlo principal. Todos os motivos anteriores foram responsáveis pela implementação do sistema distribuído, outros factores importantes foram o factor preço e o factor fiabilidade.

O sistema de comando aplicado surge da análise ao sistema desenvolvido no âmbito do projecto de final de curso Licenciatura Eng. Electrónica e Telecomunicações onde foi feita a adaptação para um sistema distribuído.

Como podemos observar na Figura 5.3 a estrutura é composta por um grupo de PLC's<sup>4</sup>, por um computador industrial, um painel de comando e um monitor de interface gráfico. Numa avaliação de custos efectuada foi verificado que era mais económico para o mesmo número de sinais entrada saída uma solução em PLCs que em placas para encaixe em barramento PCI, outra avaliação feita foi para o sistema de contagem dos eixos que necessitam de um sistema de contagem por *hardware* em que para o número de eixos que temos (dez) sai mais económico contagem por PLCs. Outro factor avaliado foi o nível de volatilidade dos dois sistemas, o sistema PLC que o sistema cartas de contagem ou de entrada saída.

#### Composição do Sistema de Comando CNC Distribuído

##### 1. COMPUTADOR INDUSTRIAL

---

<sup>2</sup>comando numérico computarizado

<sup>3</sup>controlo numérico

<sup>4</sup>autómatos programáveis ou do inglês (Programmable Logic Controllers)

- Ecra de 12.1" Touch Screen resolução 800x600 32 bits
- Processador Centrino 1.3 GHz
- 256 MB de Memória RAM
- 1 GB de Memória de Armazenamento
- Ms Windows CE
- Suporte de Rede

## 2. ENTRADAS

- PLC MASTER - Autómató Principal
- PLC SLAVE1 - Autómató Secundario1
- PLC SLAVE2 - Autómató Secundario2
- PLC SLAVE3 - Autómató Secundario3
- PLC SLAVE4 - Autómató Secundario4

## 3. SAIDAS

- PLC MASTER - Autómató Principal
- PLC SLAVE1 - Autómató Secundario1
- PLC SLAVE2 - Autómató Secundario2
- PLC SLAVE3 - Autómató Secundario3
- PLC SLAVE4 - Autómató Secundario4

## 4. EIXOS

- PLC MASTER - Autómató Principal
- PLC SLAVE1 - Autómató Secundario1
- PLC SLAVE2 - Autómató Secundario2
- PLC SLAVE3 - Autómató Secundario3
- PLC SLAVE4 - Autómató Secundario4

## 5. REDE DE COMUNICAÇÃO RS485

- PLC MASTER - Autómató Principal
- PLC SLAVE1 - Autómató Secundario1

- PLC SLAVE2 - Autómatos Secundário2
- PLC SLAVE3 - Autómatos Secundário3
- PLC SLAVE4 - Autómatos Secundário4

## 6. COMUNICAÇÃO RS232

- PLC MASTER - Autómatos Principal
- PC INDUSTRIAL

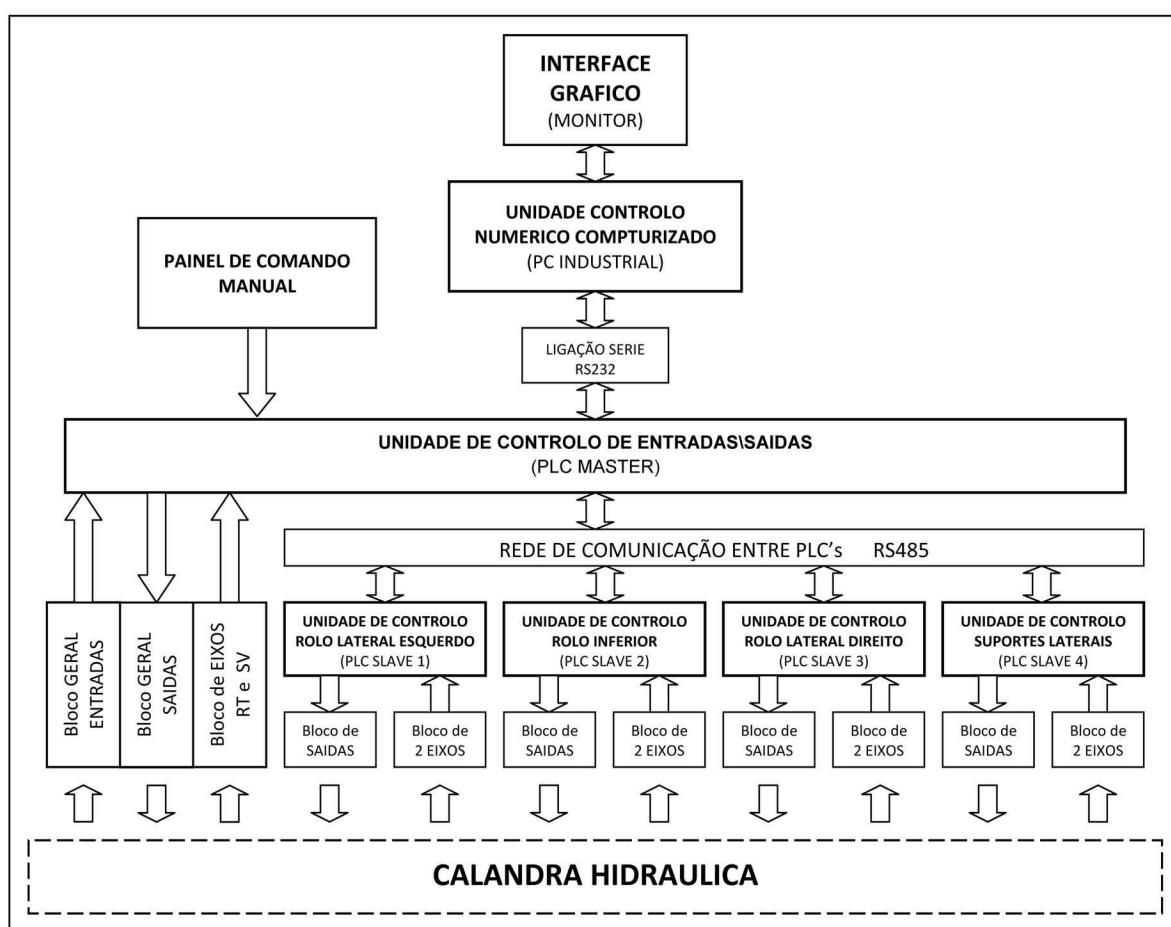


Figura 5.3: Diagrama Blocos Sistema Controlo Distribuído CNC

O sistema está sendo construído de modo que os PLCs SLAVE (Secundários) detêm o controlo do posicionamento de dois eixos que equivale a termos um PLC SLAVE (secundário) por cada rolo, visto que cada rolo possui dois eixos, um em cada extremidade do rolo, o PLC MASTER (Principal) está responsável por dois eixos, o eixo da rotação e o eixo do suporte vertical, outra função do PLC MASTER é indicar as cotas de posicionamento dos rolos

aos PLCs SLAVE respectivos, esta comunicação entre os PLCs SLAVE e o PLC MASTER é efectuada por uma rede de comunicação instalada entre todos os PLCs. A comunicação entre o nível dos actuadores (PLCs) o nível de calculo (PC Industrial) é assegurado por uma ligação série usando o protocolo RS232 entre o PLC MASTER e o PC Industrial.

O PC industrial monitoriza toda a informação do sistema, quando necessita de informar o sistema de novas cotas de posicionamento ele irá comunicar ao PLC MASTER colocando os valores das novas cotas em registos da rede de comunicação entre os PLCs, alojados no PLC MASTER, e assim cada PLC SLAVE tem acesso a toda informação de posicionamento. A informação do posicionamento actual de cada rolo é lida por cada PLC SLAVE é partilhada na rede de comunicação onde o PLC MASTER têm acesso e disponibiliza essa informação ao PC Industrial, que a projecta no monitor para que o operador da maquina tenha a informação do posicionamento dos rolos.

O PLC MASTER recebe directamente os sinais de comando manual da máquina por intermédio da ligação ao painel de comando manual que está composto por manípulos e botões para accionar a máquina em modo manual. Este funcionamento é permitido pela escolha do modo de trabalho no PC industrial ou então pela ausência do sinal do PC industrial o que permite accionar a maquina sem o sistema de controlo numérico computarizado.



Figura 5.4: Calandra equipada com Sistema de Comando Distribuído CNC

### 5.4.2 Estrutura de *Hardware* do Sistema de Comando NC

No sistema NC é retirado da estrutura o PC Industrial e o respectivo monitor que têm um peso muito significativo no desempenho da máquina e também um peso significativo no preço da solução. É colocado para interface Homem-máquina uma consola, onde é possível observar o posicionamento actual dos rolos e também permite a introdução de alguns dados para que se possa efectuar um controlo menos suportado em cálculos mas que permite ao operário algumas operações automáticas. Devido á ausência da unidade de processamento de calculo, este comando esta preparado para introduzirmos as cotas de posicionamento n a consola gráfica e assim esta informa o PLC MASTER do posicionamento pretendido, este ultimo ira informar os PLC SLAVES por intermédio da rede de comunicação da cota destino para cada rolo, os PLCs SLAVE iram colocar os rolos nas cotas pretendidas. Como este comando so realiza controlo numérico, normalmente só é implementado quatro eixos que permite o posicionamento dos rolos laterais. Como podemos observar a estrutura base é a mesma, mas existe a ausência de algumas partes que constituem o comando CNC.

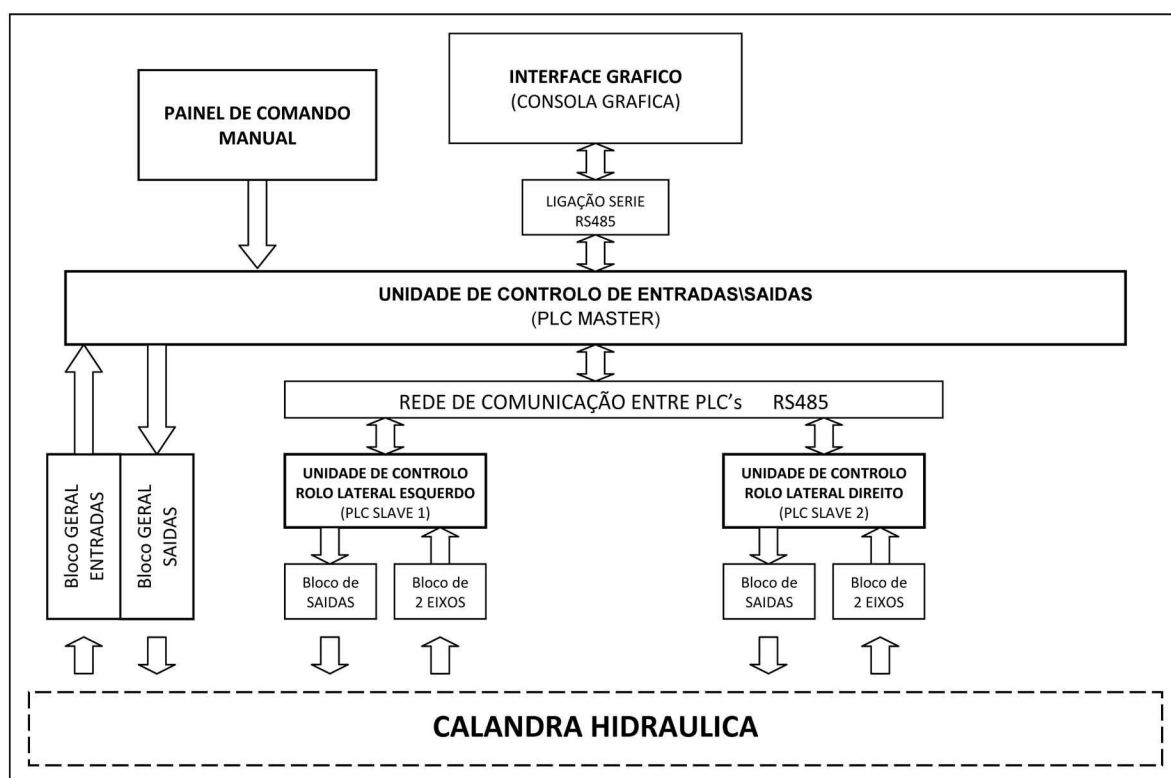


Figura 5.5: Diagrama Blocos Sistema Controlo Distribuído NC

# Capítulo 6

## *SOFTWARE CNC*

Neste capítulo vou apresentar o software desenvolvido para o sistema de comando CNC.

Como na implementação da estrutura de *hardware* também no desenvolvimento do software foi iniciado a partir do ponto de paragem do projecto de final de curso de Licenciatura e Eng. Electrónica e Telecomunicações no ano lectivo 2005/06. A evolução do sistema de comando da calandra encontra-se na terceira fase, lembrando a primeira fase desenvolvida pela empresa Marcovil,S.A onde o *software* de comando foi uma aplicação desenvolvida na plataforma *MS Dos 6.22*.

Na segunda fase de desenvolvimento (projecto de final de curso de Licenciatura e Eng. Electrónica e Telecomunicações no ano lectivo 2005/06) foi determinado implementar uma nova plataforma de sistema operativo. Esta mudança tinha como objectivo eliminar as limitações do sistema operativo *MS Dos 6.22* de modo a permitir que a nova aplicação melhora-se o interface gráfico, suporta-se um sistema multitarefa e permiti-se um acréscimo de desenvolvimento geral para aplicação final.

Tendo em vista a utilização do sistema de comando, uma das componentes principais será a interface de comunicação entre o software e o utilizador. É necessário desenvolver um "GUI" *Graphical User Interface* do tipo "user friendly", para desenvolver este interface foi necessário definir uma linguagem de desenvolvimento que permiti-se criar a aplicação com as características referidas anteriormente.

Analisando as varias linguagens e tendo como objectivo implementar novos serviços no futuro optei por continuar o desenvolvimento da aplicação de software principal na linguagem *Visual C #*, pelo facto de esta linguagem permitir o desenvolvimento de aplicações para o sistema operativo *MS Windows CE* foi um factor fundamental, outros factores para a escolha da linguagem de programação foi a compatibilidade do *Visual C #* com os outros requisitos da aplicação a construir, a compatibilidade para desenvolver uma aplicação do



tipo "GUI" *Graphical User Interface*, permitir o suporte para vários protocolos de comunicação, compatibilidade com *.NET Compact Framework* disponível no sistema operativo *MS Windows CE 5.0*.

Como existiram alterações na estrutura de *Hardware* inicialmente projectado, também o software sofreu alterações no seu planeamento no uso dos recursos.

Refiro que inicialmente estava previsto o *MS Windows XP Embedded* para sistema operativo do PC Industrial, mas com o decorrer dos trabalhos de implementação deste sistema apresentado decidi usar como sistema operativo o *Windows CE 5.0*, esta escolha tem o objectivo de caminhar para uma estrutura de *Software* que permita aplicações de tempo real, outros factores foram importantes para esta escolha, poderemos observar os mais importantes abaixo referenciados.

#### **Factores que ponderaram sobre a escolha do Sistema Operativo:**

- Não ser possível ter *timers* dentro da aplicação desenvolvida mais rápidos que 10ms no *MS Windows XP Embedded*.
- Diminuir a probabilidade de erro em consequência da diminuição do peso do sistema operativo.
- Menor exigência de recursos de processador por parte do *Windows CE 5.0*.
- Diminuir a existência de processos concorrentes ao da aplicação principal do sistema de controlo.
- Permitir introduzir um sistema operativo que suporta-se rotinas de tempo real para futuras aplicações, possível no *Windows CE 5.0*.
- O *Windows CE 5.0* tem tempos de *startup* muito menores que o *Windows XP Embedded*.
- O *Windows CE 5.0* não necessita de tempo para encerrar (processo automático com a falta de energia sem gerar erros).

Com a alteração do sistema operativo de *Windows XP Embedded* para *Windows CE 5.0* foi necessário alterar a plataforma do sistema operativo da aplicação principal do sistema de controlo. A primeira aplicação foi desenvolvida para plataforma do sistema operativo



*MS Windows XP Embedded* que não é compatível com o sistema operativo *Windows CE 5.0*.

O *software* desenvolvido no âmbito deste trabalho foi:

1. **SOFTWARE PRINCIPAL - CE CNC**

- FUNÇÃO - Controlo Numérico Principal
- INTERFACE - Sistema "GUI"
- COMUNICAÇÃO - SERIE RS232
- DESENVOLVIDO - Visual Studio 2005 C #"
- API - .NET"
- SISTEMA OPERATIVO - WINDOWS CE 5.0 .NET

2. **SOFTWARE PLC MASTER - MASTER**

- FUNÇÃO - Controlo Entradas/Saidas e Gestão Rede Comunicação RS485 entre todos os PLC's
- COMUNICAÇÃO - SERIE RS232 com PC Industrial e SERIE RS485 com PLCs
- DESENVOLVIDO - GX IEC DEveloper 7.00

3. **SOFTWARE PLC SLAVE1 - SLAVE1**

- FUNÇÃO - Controlo EIXO 1E e EIXO 2E
- COMUNICAÇÃO - SERIE RS485 REDE de PLCs
- DESENVOLVIDO - GX IEC DEveloper 7.00

4. **SOFTWARE PLC SLAVE2 - SLAVE2**

- FUNÇÃO - Controlo EIXO 5I e EIXO 6I
- COMUNICAÇÃO - SERIE RS485 REDE de PLCs
- DESENVOLVIDO - GX IEC DEveloper 7.00

5. **SOFTWARE PLC SLAVE3 - SLAVE3**

- FUNÇÃO - Controlo EIXO 3D e EIXO 4D
- COMUNICAÇÃO - SERIE RS485 REDE de PLCs
- DESENVOLVIDO - GX IEC DEveloper 7.00

6. **SOFTWARE PLC SLAVE4 - SLAVE4**

- FUNÇÃO - Controlo EIXO SD e EIXO SE
- COMUNICAÇÃO - SERIE RS485 REDE de PLCs
- DESENVOLVIDO - GX IEC DEveloper 7.00

## 6.1 Ferramentas de *Software* Utilizadas

No desenvolvimento das varias aplicações de *software* foram usadas como ferramentas de base de desenvolvimento o *Visual Studio 2005* para desenvolvimento da aplicação principal *CNC\_CE.exe* visualizado na Figura 6.1 . O *Ms Windows Xp Profissionnal* foi o sistema operativo usado para suporte do *Ms Visual Studio 2005* onde foi desenvolvida a aplicação *CNC\_CE.exe*. Para efeitos de simulação de testes da aplicação *CNC\_CE.exe* foi instalado no *Ms Windows Xp Profissionnal* um emulador do *Ms Windows CE 5.0*.

Para desenvolvimento do software aplicado nos PLCs foi usado o programa GX IEC Developer 7.00 distribuido pela Mitsubishi, neste programa foi desenvolvido todo o código aplicado aos PLCs podemos visualizar na Figura 6.2, a construção do *software* foi baseada na linguagem de contactos (*Ladder Diagram*) e texto estruturado (*Structured Text*). Este programa usou a plataforma base do *Ms Windows XP Profissionnal*.

## 6.2 Programa Principal *CNC\_CE.exe*

A aplicação de software principal (*CNC\_CE.exe*) foi desenvolvida no *Microsoft Visual Studio 2005* com base na linguagem *Visual C #*. A aplicação *CNC\_CE.exe* foi construída deforma a ser compatível com o sistema operativo *Microsoft Windows CE 5.0*. Foi usado para suporte de todas as funções do programa *CNC\_CE.exe* as classes disponíveis na plataforma de software *.NET Compact Framework*.

O *Microsoft .NET Compact Framework* é um subset do *Microsoft .NET Framework*, criado para possibilitar o desenvolvimento de aplicações para dispositivos inteligentes (*smart devices*) da mesma forma com se cria aplicativos para *Windows desktop*, ou seja aplicativos para sistemas operativos do tipo *Ms Windows Xp*. A plataforma de software *.NET Compact Framework* é um dos pacotes constituintes do sistema operativo *Ms Windows CE 5.0*, possibilitando assim o desenvolvimento da aplicação *CNC\_CE.exe* em linguagem comum, existindo algumas limitações nos recursos disponibilizados. As limitações na criação de uma aplicação para *Windows CE* existem pelo facto de ser um sistema operativo com

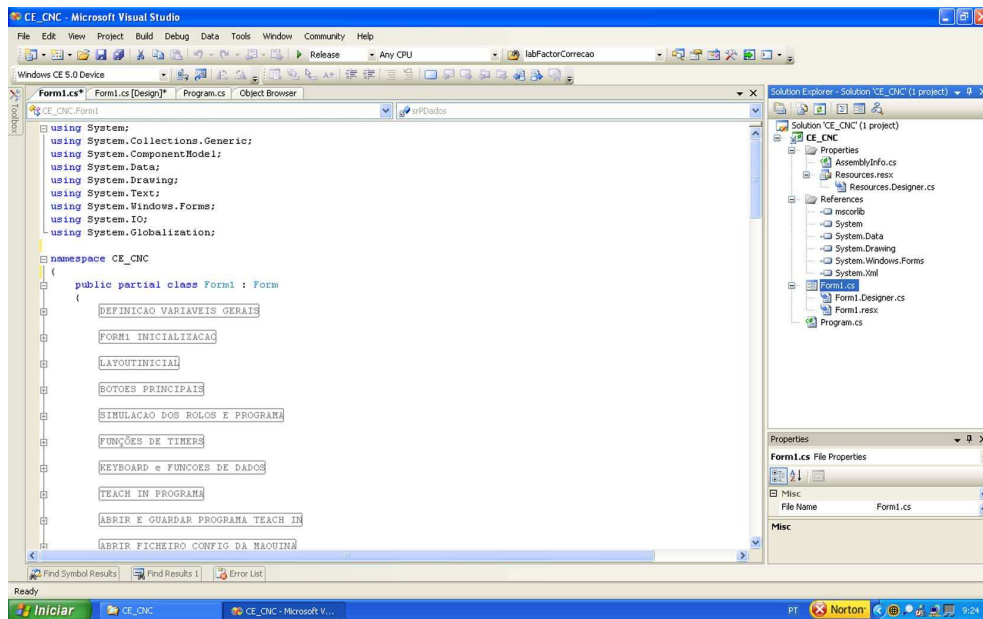


Figura 6.1: Imagem do Programa Visual Studio 2005

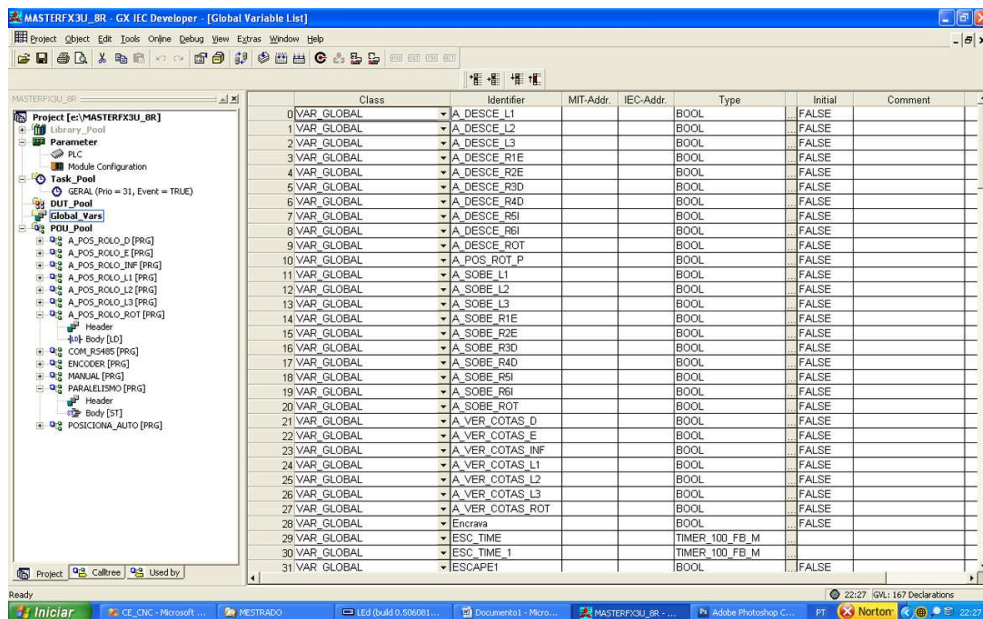


Figura 6.2: Imagem do programa GX IEC Developer 7.00

menos recursos que o sistema operativo *Ms Windows XP*, outro factor é a *.NET Compact Framework*, por ser um subconjunto da *.NET Framework* disponível no sistema operativo *Ms Windows Xp*.

Todos os códigos desenvolvidos para serem executados sob a *.NET Compact Framework* são executados por um compilador de alta performance, denominado Compilador JIT (*Just In Time Compiler*). Este compilador é responsável por otimizar o código para

o dispositivo onde o software será executado. Isto é: O mesmo software é otimizado para ser executado em dispositivos que possuem recursos de hardware ou software diferentes.

Por baixo do Compilador JIT, a CLR (*Common Language Runtime*) juntamente com uma colecção de classes (*Class Library*) permite uma gerência de processos otimizados para ser executado em equipamentos específicos. Pensando nesses dispositivos, a CLR (*Common Language Runtime*) foi otimizada para que permitisse que tudo o que era possível ser feito com a *.NET Framework*, também fosse possível de ser feito com a *.NET Compact Framework*, sendo executada nos mais diversos e diferentes equipamentos.

Na aplicação de software principal (*CNC\_CE.exe*) foi desenvolvida a "GUI" *Graphical User Interface* composta por cinco *menus* principais onde é possível ao utilizador aceder a todas as potencialidades do sistema de comando. Par de trás da interface esta implementado o algoritmo de calculo de posicionamento (**Segundo Algoritmo de Posicionamento**) apresentado no capítulo 4.

Foi desenvolvido e aplicado um sistema de programação automática em função dos dados principais da chapa, para obter um formato cilíndrico onde o utilizador introduz a espessura, comprimento, numero de secções, numero de passagens, largura, comprimento de cada secção e raio de cada secção outra das potencialidades do *software* é a possibilidade de simulação gráfica do programa introduzido pelo utilizador. Para obter o posicionamento automático foi aplicado no software principal (*CNC\_CE.exe*) o processo de calandragem apresentado no capítulo 3 (**Processo de Calandragem Desenvolvido**).

o *software* principal (*CNC\_CE.exe*) é constituindo por uma comunicação série com o PLC Master baseada no protocolo de comunicação da Mitsubishi, devido á não compatibilidade do *software* de comunicação da Mitsubishi com o sistema operativo *Windows CE* foi necessario construir a comunicação com base no protocolo de comunicação da Mitsubishi.

Outra das características do programa principal é o sistema de armazenamento de cada programa. Existem dois tipos de programas, um o programa construído com base nos dados típicos da chapa o outro é um programa construído com base no posicionamento dos rolos da maquina, este ultimo é criado com base da utilização da maquina em modo manual e registando o posicionamento dos rolos de forma a que no final seja possível um posicionamento em modo automático dos rolos efectuados em modo manual.

## 6.2.1 Diagramas de Fluxo

### Programa Principal

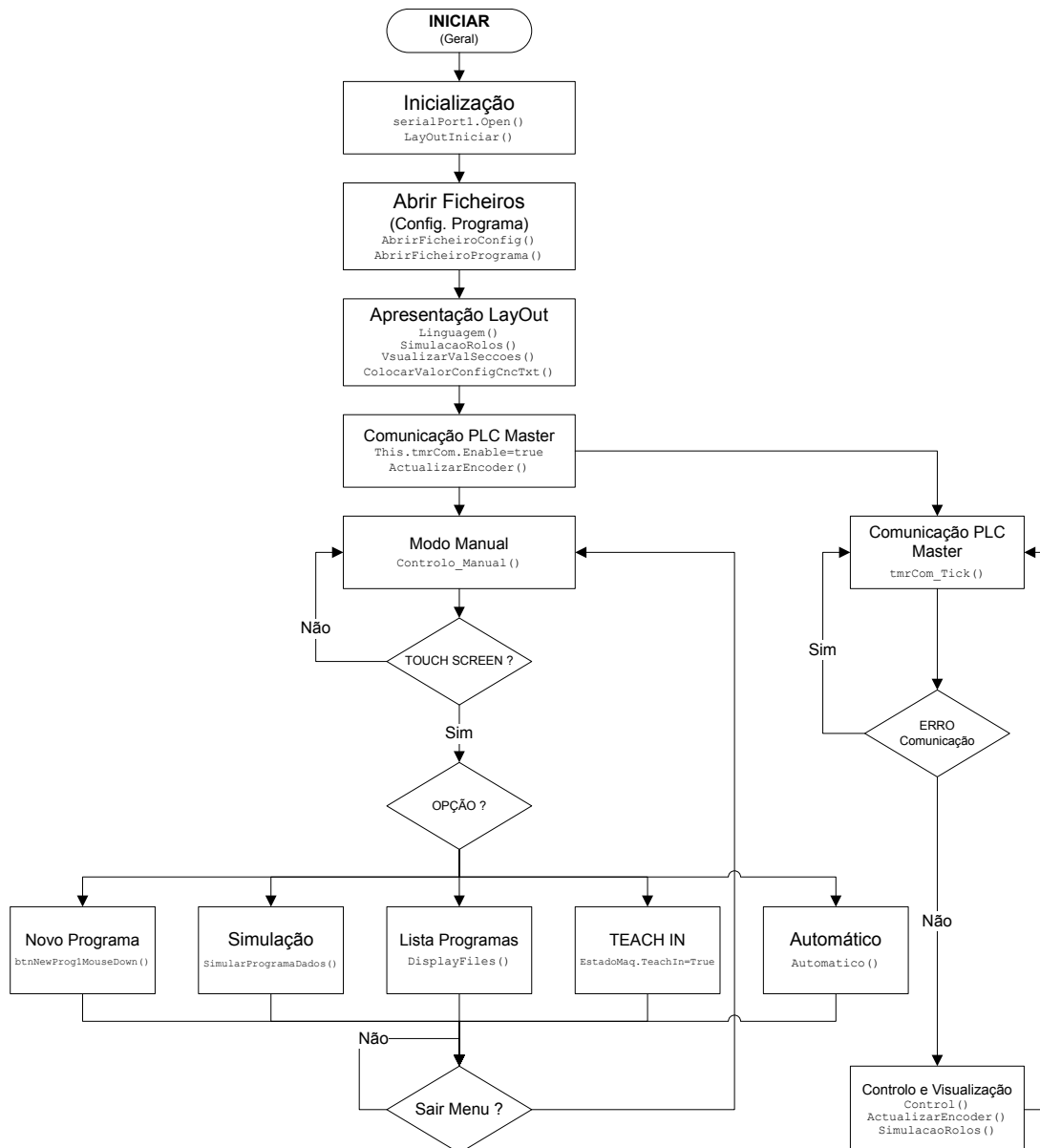


Figura 6.3: Diagrama Geral

Este diagrama demonstra o funcionamento geral do programa principal *CNC-CE.exe*. O programa está dividido em cinco menus principais.

**Primeira fase da inicialização do programa** - Inicialização de todas as estruturas e variáveis gerais do programa principal.

**Segunda fase da inicialização do programa** - Leitura dos ficheiros de *config* e do ultimo programa usado pelo operador. O ficheiro *config* é onde se encontra guardada

a informação específicos referentes a cada máquina, com a leitura de todos os dados do ficheiro *config* possibilita a configuração do programa *CNC\_CE.exe* para a máquina que está controlar, esta característica permite que a aplicação desenvolvida se aplique a todas as máquinas bastando alterar o ficheiro *config* que será específico para cada máquina.

**Terceira fase da inicialização do programa** - Verificação da linguagem do programa e actualizados todos os campos com os dados recebidos dos ficheiros abertos na fase anterior na fase final é feita a projecção gráfica do programa.

**Quarta fase da inicialização do programa** - Iniciado o processo de comunicação entre o program *CNC\_CE.exe* e o PLC Master, inicializado a função de leitura e actualização da posição dos rolos da máquina, inicialização da simulação dos rolos em função da posição real.

**Quinta fase da inicialização do programa** - Nesta fase a aplicação inicia a função de controlo estando por defeito em modo manual, todas as funções da aplicação estão em funcionamento.

Após as fases de inicialização o programa está em pleno funcionamento, como o programa é composto por cinco *menus*, existe a possibilidade de aceder a qualquer *menu* pelo acto de pressionar o botão correspondente ao *menu* pretendido. Como o monitor é do tipo *touch screen* o acesso a qualquer opção do software é muito acessível. O processo de aceder a um *menu* basta um simples toque no botão correspondente para lançar um evento, após o evento é detectado o *menu* correspondente e será activado o novo *menu* levando a alteração gráfica correspondente.

### Menu Novo Programa

O *menu* Novo Programa é a secção do programa principal que permite programar a máquina. Nesta secção é introduzido os dados de um novo programa que se pretende executar, ao iniciar esta secção é lançado o *LayOut* referente ao *menu* Novo programa e lançado um teclado de *software* que permite introduzir os valores para construir um programa.

Numa primeira fase são introduzidos os principais campos de um programa, estes campos vão surgindo de uma forma sequencial, inicializado-se no valor da Espessura passando para Comprimento, Nº Passagens, Nº Secções, Factor Correção e Largura após a introdução dos valores para os campos referidos é feita uma verificação dos dados introduzidos, se existir algum dado fora do limite permitido será enviado uma mensagem de erro e a introdução de valores regressa ao início, se for obtido sucesso nos valores introduzidos o programa passara para a introdução dos campos secundários. Os campos secundários são

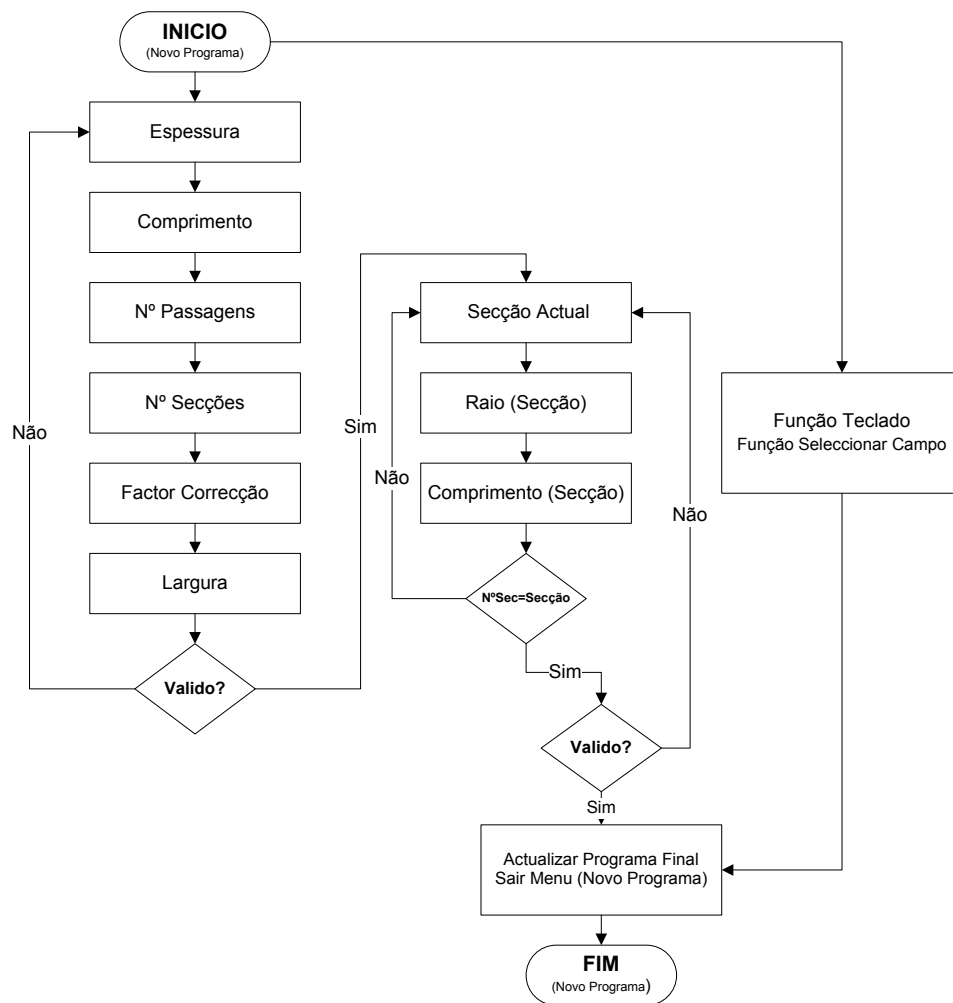


Figura 6.4: Diagrama Menu Novo Programa

referentes ao Raio e comprimento para cada secção, no fim de introduzir os campos referidos para todas as secções será validado os dados introduzidos e será feita uma simulação gráfica do programa. O programa ficará em espera enquanto o operador não pressionar no botão de sair do menu, após sair o programa retorna ao modo manual.

## Menu Simulação

No *menu* da Simulação é feita uma simulação gráfica do programa que está em memória. Após o acto de pressionar o botão da simulação surge a simulação grafica do programa actual, num *menu* lateral existe dois botões que permitem realizar a função zoom.

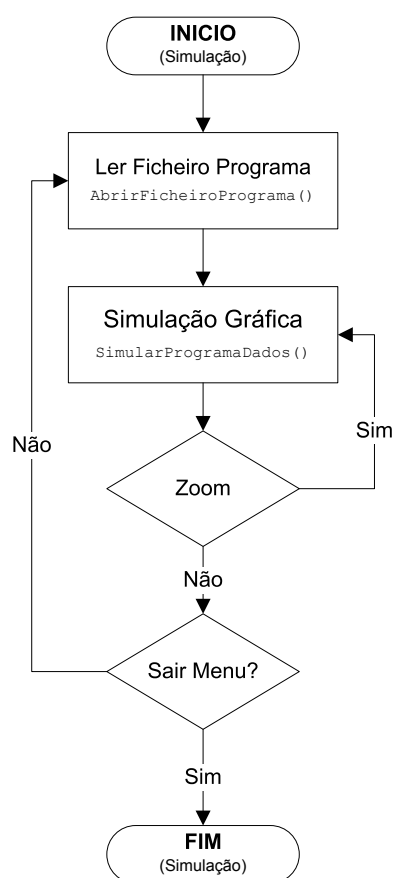


Figura 6.5: Diagrama Menu Simulação

### Menu Lista de Programas

O *menu* Lista de Programas é utilizado para guardar programas da maquina. Após o acto de pressionar o botão Lista de Programas é verificado se o modo Teach In está activo, se estiver activo será executada a listagem dos programas do tipo Teach In, se não estiver activo será feita a listagem dos programas modo automático normal. A próxima fase será escolhida pelo operador, Guardar, Apagar ou Ler um programa. Ao seleccionar um adas três opções será activado um teclado alfa-numérico que permite introduzir o nome referente ao programa.

**Opção Guardar** - Nesta opção é lançado o teclado que possibilita a introdução do nome pretendido, após a introdução do nome pressiona-se na tecla ENTER que irá validar o nome dado e chamar a função de guardar ficheiro. A função guardar ficheiro ira guardar o programa em memoria no directório de programas. Após executar a função guardar ira ser executada de novo a função listar programas, mantendo assim a listagem actualizada.

**Opção Ler** - Esta opção permite-nos executar um programa que esteja guardado. Pressionando o botão de Ler será lançado o teclado para introdução do nome do programa,



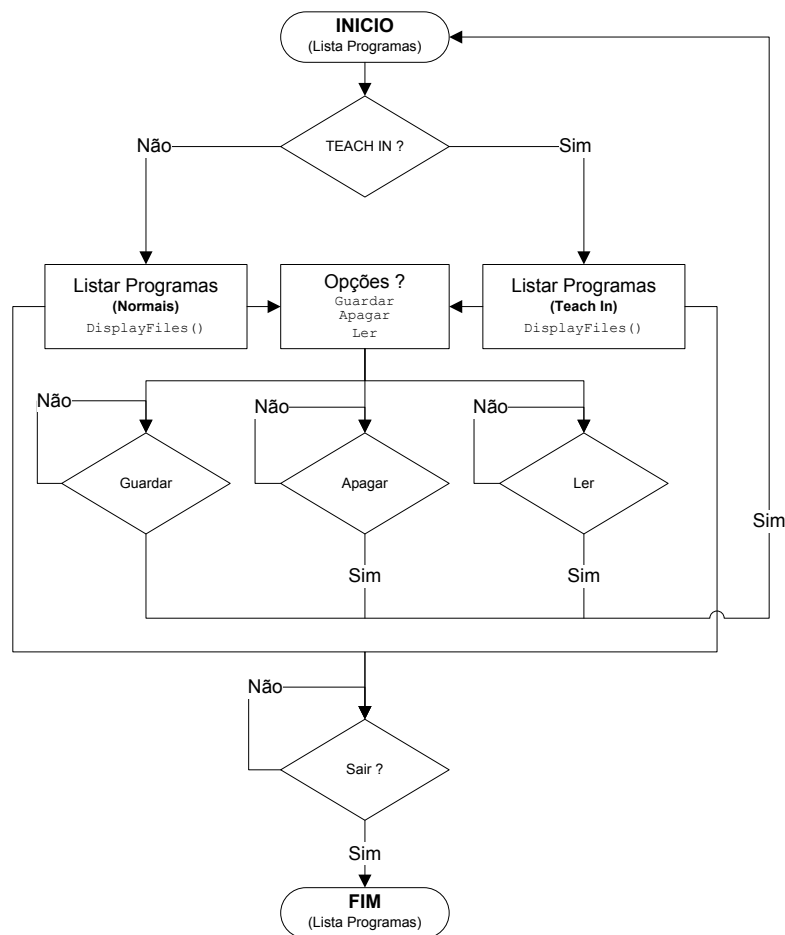


Figura 6.6: Diagrama Menu Lista de Programas

quando pressionado o botão ENTER é verificado a existência do programa. Verificado o programa é transferida toda a informação para a estrutura do programa que é executada pelo modo automático.

**Opção Apagar** - Esta opção é semelhante às anteriores diferenciando-se na execução, o lançamento do teclado é igual a parte de verificação do nome é igual, após a verificação do nome é eliminado o programa e executada a listagem de programas para actualizar a lista de programas.

Dentro do *menu* Listar Programas poderá executar varias tarefas de Guardar, Ler ou Apagar, para sair basta pressionar no botão Listar Programas, após esta operação a maquina regressa ao *menu* Base que é o *menu* do modo manual.

## Menu TEACH IN

Ao pressionar o botão TEACH IN é verificado se o modo teach in está activo e desactiva ou activa em função do estado actual. Com este modo activo permite ao utilizador criar

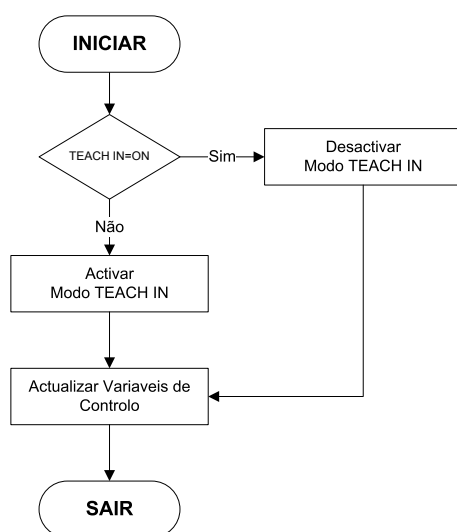


Figura 6.7: Diagram TEACH IN

um programa baseado numa execução em modo manual, este programa ficará preparado para uma execução em modo automático. O programa criado poderá ser guardado no *menu* Lista de Programas. Ao entrar no *menu* automático com o modo Teach In activo o programa que é executado é do tipo Teach In.

### Menu Automático

Na secção *menu* Automático encontra-se dois métodos de trabalho, um referente a um programa normal programado na secção Novo programa e outro referente aos programas criados em Teach In. Os programas criados no *menu* Novo Programa são executados por varias fases. A primeira fase será o posicionamento dos rolos para inserir a chapa, segunda fase apertar a chapa, terceira e quarta fase execução da calandragem para uma determinada secção, após a execução da secção é verificado se o programa cotem mais secções ou passagens, assim que se esgote o,limite de secções passagens o programa sairá das fases de calandragem e entra numa fase de correcção, nesta fase é possível fazer voltar a chapa á maquina e ajustar a curvatura. No final da execução da correcção o operador poderá iniciar nova sequência de trabalho. O Modo teach In é executado numa so fase e repetido o numero de vezes das passagens capturadas anteriormente na construção do programa.

No inicio de cada fase em que se encontra posicionando, é enviada uma trama de bits para o PLC Master contendo as posições dos rolos. O posicionamento sequencial dos rolos é activado pelo PC, iniciamos o posicionamento do rolo inferior, após o posicionamento do rolo inferior é desactivado o bit que tinha a função de activar o movimento no PLC, esse

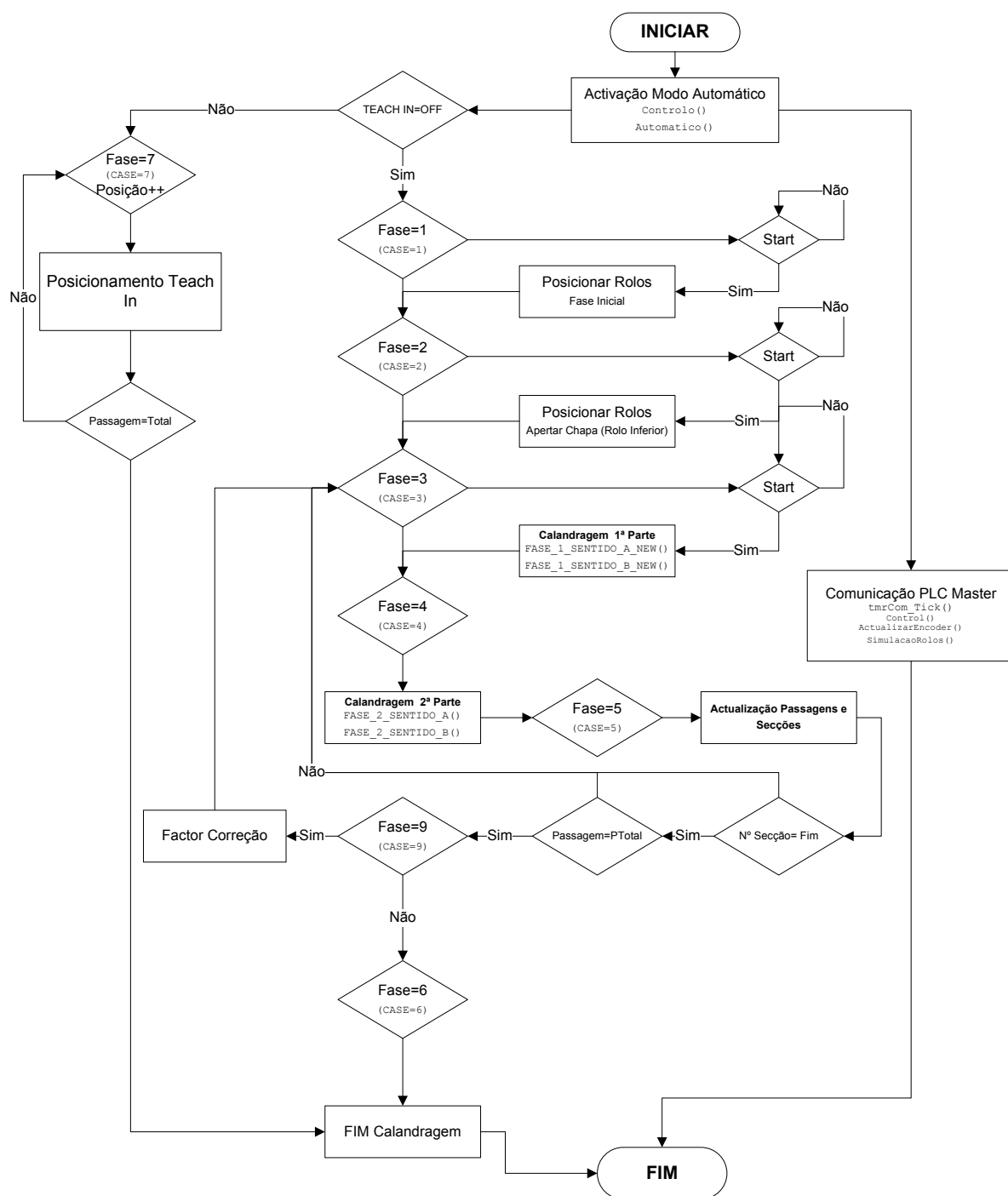


Figura 6.8: Diagrama Modo Automático

bit é recebido pelo PC que assim entende que o rolo inferior está posicionado dando origem a um novo posicionamento dos rolos laterais.

## 6.2.2 Classes do Sistema

Passamos agora para uma apresentação da estrutura de código desenvolvida na aplicação principal *CNC\_CE.exe*, será dividida em varias partes, iniciando-se na apresentação das classes bases do sistema, passando para as estruturas criadas para suporte do código e por final a apresentação das funções desenvolvidas.

### *Nomespaces* usados pelo programa da *.NET Compact Framework*

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Text;
7 using System.Windows.Forms;
8 using System.IO;
9 using System.Globalization;
```

**Código 1:** *Nomespaces* Usados da *.NET Compact Framework*

Todas as classes usadas no aplicação *CE\_CNC.exe* estão suportadas pelos *Nomespaces* apresentados no Código 1.

## 6.2.3 Estruturas Principais do Código *CNC\_CE.exe*

### public struct EstadoMaquina

```
1 public struct EstadoMaquina
2 {
3     public bool TeachIn; //Variavel Controlo Modo Teach IN ON\Off
4     public bool Automatico; //Variavel Controlo Modo Automatico\Manual
5     public bool Avanco; //Variavel Controlo Modo Avanco ON \OFF
6     public bool Cilindrico; //Variavel Controlo Modo Cilindrico\Cone
7     public bool Programa; //Variavel Controlo Menu Programa
8     public bool File; //Variavel Controlo Menu File
9     public bool Simulacao; //Variavel Controlo Menu Simulação
10    public bool Gravar; //Variavel Controlo Menu File Gravar
11    public bool Apagar; //Variavel Controlo Menu File Apagar
12    public bool Ler; //Variavel Controlo Menu File Abrir
13    public bool ConfigCNC; //Variavel Controlo Menu File Config
14    public float DistRLatEixoInfMin; //Variavel Medida Distancia Rolo L Inf
15    public float PosInicialChapaBatente; //Variavel Medida Dist. Pos. da Chapa Inicial Bat.
16    public float PosInicialChapaInferior; //Variavel Medida Dist. Pos. da Chapa Inicial Inf.
17    public float DistV_RLatMin; //Variavel Medida Distancia Vertice RLaterai Minimo
18    public bool AbreBraco; //Variavel Controlo Modo Aut. Abertura Braço ON\Off
19 }
```

**Código 2:** Estrutura *public struct EstadoMaquina*

Esta estrutura é composta por variáveis de controlo de fluxo, são variáveis que têm a informação do estado de cada menu(se está activo ou inactivo) e também dos modos de

trabalho da maquina. As variáveis que indicam distancias são referentes ao posicionamento da chapa na posição inicial.

### public struct PosicaoRolos

```

1  public struct PosicaoRolos
2  {
3      public int NPassagem;           //Variavel Controlo Numero Passagem
4      public int Posicao1E;            //Variavel Posição Cilindro Motores Esquerdo
5      public int Posicao2E;            //Variavel Posição Cilindro Frente Esquerdo
6      public int Posicao3D;            //Variavel Posição Cilindro Motores Direito
7      public int Posicao4D;            //Variavel Posição Cilindro Frente Direito
8      public int Posicao5I;            //Variavel Posição Cilindro Motores Inferior
9      public int Posicao6I;            //Variavel Posição Cilindro Frente Inferior
10     public int Rotacao;              //Variavel Rotação
11     public int PosicaoSD;             //Variavel Posição Suporte Direito
12     public int PosicaoSE;             //Variavel Posição Suporte Esquerdo
13     public int PosicaoSV;             //Variavel Posição Suporte Vertical
14     public int OffSetRot;            //Variavel OffsetRotação
15     public int RotacaoTotal;         //Variavel Posição da Chapa
16     public float AngRD;              //Variavel Angulo Inclinação Rolo Direito
17     public float AngRE;              //Variavel Angulo Inclinação Rolo Esquerdo
18     public float AngRI;              //Variavel Angulo Inclinação Rolo Inferior
19 }
20
```

**Código 3:** Estrutura *public struct PosicaoRolos*

Esta estrutura é composta por variáveis do tipo *Int 32bits* e variáveis do tipo *float*. As variáveis do tipo *Int 32bits* apresentam o valor da posição actual de cada Eixo de posicionamento existente na maquina, as variáveis do tipo *float* apresentam o valor do declive dos rolos principais da máquina em graus.

### public struct RegistoRolos

```

1  public struct RegistoRolos
2  {
3      public double Rolo1E;
4      public double Rolo2E;
5      public double Rolo3D;
6      public double Rolo4D;
7      public double Rolo5I;
8      public double Rolo6I;
9      public double Rotacao;
10     public double Suporte1;
11     public double Suporte2;
12     public double Suporte3;
13     public double Rot_offset;
14     public double Dif_Rotacao;
15 }
```

**Código 4:** Estrutura *public struct RegistosRolos*

Estrutura de registos de 32 bits usada para armazenar os valores calculados do posicionamento em modo automático.

**public struct RolosAutomato**

```
1 public struct RolosAutomato
2 {
3     public short Rolo1E;
4     public short Rolo2E;
5     public short Rolo3D;
6     public short Rolo4D;
7     public short Rolo5I;
8     public short Rolo6I;
9     public short Rotacao;
10    public short Suporte1;
11    public short Suporte2;
12    public short Suporte3;
13    public short Rot_offset;
14 }
```

**Código 5:** Estrutura *public struct RolosAutomato*

Estrutura de usada na transferência dos valores de posicionamento de cada eixo de controlo da estrutura `public struct RegistoRolos` de 32 bits para o PLC Master que usa registos de 16 bits.

**public struct ControloRolo**

```
1 public struct ControloRolo
2 {
3     public bool A_POS_E;
4     public bool A_POS_D;
5     public bool A_POS_INF;
6     public bool A_POS_ROT;
7     public bool A_POS_L1;
8     public bool A_POS_L2;
9     public bool A_POS_L3;
10    public bool A_POS_FB;
11    public bool A_POS_AB;
12    public bool ROLO_E;
13    public bool ROLO_D;
14    public bool ROLO_INF;
15    public bool ROLO_ROT;
16    public bool ROLO_L1;
17    public bool ROLO_L2;
18    public bool ROLO_L3;
19    public bool BRACO;
20 }
```

**Código 6:** Estrutura *public struct ControloRolo*

Estrutura composta por variáveis de controlo de posicionamento dos rolos da maquina. Variáveis usadas para activar o movimento dos rolos, outras para verificar o estado de posicionamento. Estas variáveis são usadas dentro da função de posicionamento automático.

**public struct DadosMaquina**

Estrutura onde é armazenado todas as medidas físicas da geometria da maquina, como diâmetros de rolos, distancia entre centros de rolos, ângulos de desvios, ângulos carac-

```

1  public struct DadosMaquina
2  {
3      public int Largura;
4      public int Espessura;
5      public int DiamRoloSuperior;
6      public int DiamRoloInferior;
7      public int DiamRoloLateral;
8      public int DistRSVertice;
9      public int DistRLatRSup;
10     public int AnguloCarac;
11     public int DesvioAngular;
12     public int NImpulsosEncoder;
13     public int NDentesRGrande;
14     public int NDentesRPequena;
15     public float RelacaoRotacao;
16     public int CursoLateral;
17     public int CursoInferior;
18     public int NImpulsosPosicao;
19     public int DesliseSubida;
20     public int DesliseDescida;
21     public int DesliseRotacao;
22     public int TempoAbertura;
23     public int TempoMaquina;
24     public int Linguagem;
25     public float AngRadMaq;
26     public float FactorRotacao;
27     public float Escala;
28     public int SUP1;
29     public int SUP2;
30     public int SUP3;
31     public int R1E;
32     public int R2E;
33     public int R3D;
34     public int R4D;
35     public int R5I;
36     public int R6I;
37     public int DistRoloSTorre;
38 }

```

**Código 7:** Estrutura *public struct DadosMaquina*

terísticos da maquina, referencia da maquina, relação de rotação, numero de impulsos dos encoders e cursos máximos de cada eixo de posicionamento.

## public struct ProgramaDados

```

1  public struct ProgramaDados
2  {
3      public float Espessura;
4      public int TComp;
5      public int NPassagens;
6      public int NSeccoes;
7      public int Largura;
8      public int FCorrecao;
9      public int FCorrecaoAnt;
10     public float[] SComp;
11     public float[] SRAio;
12 }

```

**Código 8:** Estrutura *public struct ProgramaDados*

Estrutura de suporte ao programa automático, onde é guardado os dados principais da chapa. Espessura, numero de secções, comprimento, comprimento total, largura, raio de cada secção, comprimento de cada secção e numero de secções.

**public struct ControlMaquina**

```

1  public struct ControlMaquina
2  {
3      public float Zoom;
4      public int FactorZoom;
5      public int NumSecActual;
6      public int NumPassagem;
7      public bool ACTIVO;
8      public int NumPTeachIn;
9      public int NumTPTeachIn;
10 }

```

**Código 9:** Estrutura *public struct ControlMaquina*

Estrutura composta por variáveis de controlo de funcionamento como numero de passagens de modo automático, numero de passagens do modo Teach IN e variáveis de controlo de zoom para o menu de simulação.

**public struct Mensagem**

```

1  public struct Mensagem
2  {
3      public string POS_ROLO_E;
4      public string POS_ROLO_D;
5      public string POS_ROLO_I;
6      public string POS_ROLO_L1;
7      public string POS_ROLO_L2;
8      public string POS_ROLO_L3;
9      public string POS_ROLO_ROT;
10     public string POS_ROLO_D_ROT;
11     public string POS_ROLO_E_ROT;
12     public string SOBE_ROLO_E;
13     public string DESCE_ROLO_E;
14     public string SOBE_ROLO_D;
15     public string DESCE_ROLO_D;
16     public string SOBE_ROLO_I;
17     public string DESCE_ROLO_I;
18     public string SOBE_ROLO_L1;
19     public string DESCE_ROLO_L1;
20     public string SOBE_ROLO_L2;
21     public string DESCE_ROLO_L2;
22     public string SOBE_ROLO_L3;
23     public string DESCE_ROLO_L3;
24     public string RODAR_E;
25     public string RODAR_D;
26     public string NIVELAR;
27     public string START;
28     public string FIM;
29     public string CONTINUAR;
30     public string CORRECAO;
31     public string POTENCIA;
32     public string ABRIR_BRAÇO;
33     public string FECHAR_BRAÇO;
34     public string AVANCO_OFF;
35     public string AVANCO_ON;
36     public string FORMA;
37     public string ABERTURA_ON;
38     public string ABERTURA_OFF;
39     public string MODOMANUAL;
40     public string MODOAUTOIMATICO;
41     public string LISTARTEACH;
42     public string GRAVAR;
43     public string APAGAR;
44     public string OPEN;
45 }

```

**Código 10:** Estrutura *public struct Mensagem*



Estrutura usada para suporte das mensagens emitidas pelo programa nas varias partes. Esta estrutura foi criada para possibilitar que o mesmo programa possa ser apresentado em vários idiomas.

## 6.2.4 Inicialização das Estruturas e Variáveis do Programa

```

1  public string EndCom;
2  public int COM_OK = 0;
3  public byte[] dados = new byte[80];
4  public string[] Encoder = new string[14];
5  public int[] ValEncoder = new int[14];
6  public const int VMAXTEACHIN=80;
7  public int varControlPrincipais = 0; // Var para Controlo dos Dados Principais
8  public bool TmrActive = false;
9  private int CntIndexTeach = 0;      // Variavel cnt do Teach In
10 ProgramaDados DadosP;
11 DadosMaquina DadosM;
12 public static PosicaoRolos[] PRolosTeachIn=new PosicaoRolos[VMAXTEACHIN];
13 public static PosicaoRolos PRolos = new PosicaoRolos();
14 public static RegistoRolos RegRolos = new RegistoRolos();
15 public static RolosAutomato RegFx3u = new RolosAutomato();
16 public ControlMaquina ControlCnc=new ControlMaquina();
17 public EstadoMaquina EstadoMq = new EstadoMaquina();
18 ControloloRolo CONTROL = new ControloloRolo();
19 ListView lv11 = new ListView();
20 Mensagem MSN = new Mensagem();
21 public bool Simular = true;
22                                     // Controlo de leitura porta serie
23 public bool ActivoCom = false;
24 public bool WaitCom = false;
25 public bool ValOk = false;
26 public int CntTime = 1;
27 public int CONTROLCOM = 0;
28 public int NLISTATEACH = 0;
29
30                                     // variaveis do antigo programa
31 public int VARCONTAUTO = 0;
32 public int FASE = 0;
33 public bool FASE_CONTROL = false;
34 public int auxiliar;
35 public bool CORRECAO = false;
36 public bool BTN_ON = false;      // ACTIVA SEMPRE QUE ENTRA NUM MENU
37 public bool WRITEFX3U = false;   // ACTIVA QUANDO E PARA TRANSMITIR PARA O AUTOM.
38 public string Address;           // ENDERESS DO AUTOMATO
39 public int FX3U = 0;              // FX3U=0 NADA // FX3U=1 SET BIT // FX3U=2 MAN/AUTO // FX3U=3 AUTO/MAN
40 public int NumCom = 0;           // Numero de Comunicacao dentro do ciculo
41 public bool VALACTUAL = false;
42 public int AUTO = 0;
43 public bool RESET = false;
44 public int TempMin=0;
45 public int TempoHoras=15;
46 public bool SEC = false;
47 public bool MIN = false;
48 public float esp_perdae=0;
49 public float esp_perdae1=0;
50 public float esp_perdad=0;
51 public float esp_perdad1=0;
52 public int SENTIDO_A;
53 public bool PASS = false;
54 public bool ERRO = false;
55 public bool ERRO1 = false;
56 public bool ERRO2 = false;
57 public bool ERRO3 = false;
58 public bool ERRO4 = false;
59 public bool ACT_CORRECAO = false;
60
61 public float ESCALA_PICEL;
62 int Xlat=0;
63 int Ylat=0;
64 public double Angulo;
65 public int PosXl11=0;
66 public int PosYl11=0;
67 public int PosXl21=0;
68 public int PosYl21=0;

```

```
1
2
3
4 public int VarTeachKeyBoard = 0;
5 public int indiceT=0;
6 public bool EditarTeach = false;
7 public float RotacaoTotal;
8 public int NPassagemInicial = 0;
9 public int FatorCorrecaoInicial = 0;
10 public bool EditTeahIn = false;
11 public double PosOposto = 0;
12 public double PosOposto1 = 0;
```

**Código 11:** Estrutura *Inicialização das Estruturas e Variáveis do Programa*

Aqui é inicializado todas as estruturas apresentadas anteriormente, são também definidas e inicializadas as restantes variáveis gerais do programa.

Todas as variáveis e estruturas inicializadas pertencem a classe *public partial class Form1* que é a classe principal do programa *CNC\_CE.exe*.

### 6.2.5 Funções Principais do Programa *CNC\_CE.exe*

Passamos a apresentar nesta secção todas as funções importantes desenvolvidas que pertencem ao programa *CNC\_CE.exe*. O programa foi desenvolvido em por partes, inicialmente foi implementado o visual grafico. Numa segunda fase foi desenvolvido o interface entre menus, e por final foram desenvolvidas funções específicas para completar o programa de modo a executar todas as funções de trabalho que foram determinadas.

#### **private void LayOutInicial()**

Na inicialização do programa é chamada a função `private void LayOutInicial()` que serve de suporte ao iniciar do programa. Esta função é usada para inicializar a comunicação entre o computador industrial e o PLC Master, outra das funções é a leitura do ficheiro de configuração da maquina onde está guardada as medidas específicas de cada maquina. São também executadas as funções que contribuem para a configuração do programa, como a função de definição da linguagem.

```
1 private void LayOutInicial()
2 {
3     PRolos.Posicao1E = 0;
4     PRolos.Posicao2E = 0;
5     PRolos.Posicao3D = 0;
6     PRolos.Posicao4D = 0;
7     PRolos.Posicao5I = 0;
8     PRolos.Posicao6I = 0;
9
10    AbrirFicheiroConfig("\\System\\CNC\\config.txt");
11    this.label3.Text = DadosM.Largura.ToString() + "/" + DadosM.Espessura.ToString();
12    AbrirFicheiroPrograma("\\System\\CNC\\dados.txt");
13    VisualizarValSeccoes(DadosP.NSeccoes);
14    ColocarValorConfigCncTxt();
15
16    try
17    {
18        this.tmrCom.Enabled = true;
19    }
20    catch
21    {
22        labMsnPrincipal.Text = "ERRO DE COMUNICACAO 2";
23    }
24    DadosM.RelacaoRotacao = (float)((float)(Math.PI * DadosM.DiamRoloSuperior * 1.0f) /
25    (float)((float)(DadosM.NDentesRGrande * 1.0f / DadosM.NDentesRPequena * 1.0f) *
26    (float)DadosM.NImpulsosEncoder * 1.0f));
27
28    DadosM.AngRadMaq = (float)(this.DadosM.AnguloCarac * Math.PI / 180.0f);
29    ControlCnc.Zoom = 1;
30    ESCALA_PICEL = (int)(DadosM.DiamRoloSuperior / 63.5);
31    SimulacaoRolos();
32    Linguagem();
33    PosRoloInicial();
34
35    EstadoMaq.Avanco = false;
36    this.panAvanco.BackColor = Color.Linen;
37    this.labAvanco.BackColor = Color.Linen;
38    this.labAvanco.Text = MSN.AVANCO_OFF;
39
40    EstadoMaq.AbreBraco = false;
41    this.labAbertura1.Text = MSN.ABERTURA_OFF;
42    this.panAbertura.BackColor = Color.Linen;
43    this.labAbertura1.BackColor = Color.Linen;
44 }
45 #endregion
```

**Código 12:** Função *Função de Inicialização do Programa*

## public void SimulacaoRolos()

```

1  public void SimulacaoRolos()          // Simulação dos Rolos no Ecra
2  {
3
4      this.Refresh();
5      Pen pen_preto = new Pen(Color.White, 1);
6      Pen pen_preto_fino = new Pen(Color.Black, 2);
7      SolidBrush rolosBrush = new SolidBrush(Color.Red);
8      SolidBrush rolosLatBrush = new SolidBrush(Color.Green);
9
10     //COORDENADAS DO ROLO SUPERIOR
11
12     int Cx= 370;
13     int Cy= 265;
14     int Cix = 370;
15
16     int DRs = Convert.ToInt16(DadosM.DiamRoloSuperior / ESCALA_PICEL);
17     int DRi = Convert.ToInt16(DadosM.DiamRoloInferior / ESCALA_PICEL);
18     int DRl = Convert.ToInt16(DadosM.DiamRoloLateral / ESCALA_PICEL);
19     int RRs=Convert.ToInt16(DadosM.DiamRoloSuperior / (ESCALA_PICEL*2));
20     int RRi=Convert.ToInt16(DadosM.DiamRoloInferior / (ESCALA_PICEL*2));
21     int RRl = Convert.ToInt16(DadosM.DiamRoloLateral / (ESCALA_PICEL * 2));
22     int Ciy = Cy + RRs + RRi;
23     int Cl1x = Cx - Xlat;
24     int Cl2x = Cx + Xlat;
25     int Cl1y = Cy + Ylat;
26     int Cl2y = Cy + Ylat;
27
28     int PosXl1=Convert.ToInt16((PRolos.Posicao2E / (ESCALA_PICEL * 8)) * Math.Sin(Angulo));
29     int PosYl1=Convert.ToInt16((PRolos.Posicao2E / (ESCALA_PICEL * 8)) * Math.Cos(Angulo));
30     int PosXl2=Convert.ToInt16((PRolos.Posicao4D / (ESCALA_PICEL * 8)) * Math.Sin(Angulo));
31     int PosYl2 = Convert.ToInt16((PRolos.Posicao4D / (ESCALA_PICEL * 8)) * Math.Cos(Angulo));
32
33     if((PosXl1!=PosXl11)&&(PosYl1!=PosYl11))
34     {
35         PosXl11 = PosXl1;
36         PosYl11 = PosYl1;
37     }
38     if ((PosXl2 != PosXl21) && (PosYl2 != PosYl21))
39     {
40         PosXl21 = PosXl2;
41         PosYl21 = PosYl2;
42     }
43
44     Graphics g = this.CreateGraphics();
45     g.FillEllipse(rolosBrush, Cx - RRs, Cy - RRs, DRs, DRs);
46     g.DrawEllipse(pen_preto, Cx - RRs, Cy - RRs, DRs, DRs);
47
48     g.FillEllipse(rolosBrush, Cix - RRi, (Ciy - RRi) + Convert.ToInt16(PRolos.Posicao6I /
49         (ESCALA_PICEL * 8)), DRi, DRi);
50     g.DrawEllipse(pen_preto, Cix - RRi, (Ciy - RRi) + Convert.ToInt16(PRolos.Posicao6I /
51         (ESCALA_PICEL * 8)), DRi, DRi);
52
53     g.FillEllipse(rolosBrush, (Cl1x - RRl) - PosXl11, (Cl1y - RRl) + PosYl11, DRl, DRl);
54     g.DrawEllipse(pen_preto, (Cl1x - RRl) - PosXl11, (Cl1y - RRl) + PosYl11, DRl, DRl);
55
56     g.FillEllipse(rolosBrush, (Cl2x - RRl) + PosXl21, (Cl2y - RRl) + PosYl21, DRl, DRl);
57     g.DrawEllipse(pen_preto, (Cl2x - RRl) + PosXl21, (Cl2y - RRl) + PosYl21, DRl, DRl);
58
59     if (DadosM.SUP1 != 0)
60     {
61         g.FillEllipse(rolosBrush, 145, 300, 28, 28);
62         g.DrawEllipse(pen_preto, 145, 300, 28, 28);
63     }
64     if (DadosM.SUP2 != 0)
65     {
66         g.FillEllipse(rolosBrush, 355, 120, 26, 26);
67         g.DrawEllipse(pen_preto, 355, 120, 26, 26);
68     }
69     if (DadosM.SUP3 != 0)
70     {
71         g.FillEllipse(rolosBrush, 560, 300, 28, 28);
72         g.DrawEllipse(pen_preto, 560, 300, 28, 28);
73     }
74 }

```

**Código 13:** Função *Função de SimulacaoRolos*

A função `public void SimulacaoRolos()` é usada para simular graficamente o posicionamento dos rolos da calandra. Esta função é executada no início do programa e depois

é executada periodicamente para manter a projecção gráfica do posicionamento dos rolos da máquina.

Esta função usa a classe gráfica do sistema *using System.Drawing* para desenhar círculos, a projecção gráfica no monitor está relacionada com a geometria da máquina, os círculos que simulam os rolos da máquina deslocam-se à escala mantendo uma relação gráfica com o posicionamento dos rolos da máquina. O posicionamento dos círculos estão em função da posição real dos rolos.

### struct ArcoCentro

```

1  struct ArcoCentro
2  {
3      public int Cx;
4      public int Cy;
5      public int Xfinal;
6      public int Yfinal;
7
8      public int NGraus;
9      public int Angulo;
10
11     public int End;
12     public int Start;
13     public int RaioT;
14
15     public ArcoCentro(int Xinicial, int Yinicial, int StartAngulo, int Raio, int comprimento)
16     {
17         Start = (int)(StartAngulo / 2);
18         RaioT = Raio;
19         Angulo = Convert.ToInt32(((360 * comprimento) / (2 * Math.PI * Raio))/2);
20         End = Start + Angulo;
21         NGraus = Math.Abs((End - Start) + 1);
22
23         Cy = Convert.ToInt32(((Yinicial - Raio * Math.Sin(Start * (Math.PI / 90F) + (Math.PI / 2)))));
24         Cx = Convert.ToInt32(((Xinicial - Raio * Math.Cos(Start * (Math.PI / 90F) + (Math.PI / 2)))));
25
26         Xfinal = (int)Math.Round(Cx + Raio * (Math.Cos(End * (Math.PI / 90F) + (Math.PI / 2))));
27         Yfinal = (int)Math.Round(Cy + Raio * (Math.Sin(End * (Math.PI / 90F) + (Math.PI / 2))));
28     }
29 }
```

**Código 14:** Estrutura *struct ArcoCentro*

Para simulação gráfica de um programa foi necessário construir algumas ferramentas de modo a criar um conjunto de pontos com o formato do programa inserido na máquina. Como um programa é constituído por uma secção ou mais secções cilíndricas onde é necessário unir secção a secção de forma a que no final se obtenha o conjunto de todas as secções, para auxiliar a união das secções foi criada uma estrutura onde é calculado o centro da secção e ângulo de entrada e saída, comprimento e ângulo de abertura. Com auxílio da estrutura **struct ArcoCentro** e feita a projecção gráfica do formato do programa inserido na máquina.

### private void SimularProgramaDados()

A Função **private void SimularProgramaDados()** é a função responsável pela projecção gráfica do programa inserido na máquina. Esta função inicializa a projecção gráfica

```

1 private void SimularProgramaDados()
2 {
3     // SIMULAÇÃO DOS ROLOS
4     this.Refresh();
5     Pen LinhaBranca = new Pen(Color.White, 1);
6     Pen LinhaRed = new Pen(Color.Red, 2);
7     Pen LinhaPreta = new Pen(Color.Black, 2);
8     Pen LinhaAmarelo = new Pen(Color.Lime, 2);
9     SolidBrush RolosColor = new SolidBrush(Color.Red);
10    Graphics g = this.CreateGraphics();
11
12    int i = 0;
13    int Cnt;
14    int ri_y = 420;
15    int ri_x = 370;
16    int Xfinal;
17    int Yfinal;
18    int XfinalA;
19    int YfinalA;
20    int End;
21
22    // COORDENADAS DO ROLO SUPERIOR
23    g.FillEllipse(RolosColor, (int)((ri_x) - (int)(27 / ControlCnc.Zoom)), (int)(ri_y - (56 /
24        ControlCnc.Zoom) - (int)((20) / (16 * ControlCnc.Zoom))), (int)(54 / ControlCnc.Zoom),
25        (int)(54 / ControlCnc.Zoom));
26    g.DrawEllipse(LinhaBranca, (int)((ri_x) - (int)(27 / ControlCnc.Zoom)), (int)(ri_y - (56 /
27        ControlCnc.Zoom) - (int)((20) / (16 * ControlCnc.Zoom))), (int)(54 / ControlCnc.Zoom),
28        (int)(54 / ControlCnc.Zoom));
29
30    //COORDENADAS DO ROLO INFERIOR
31    g.FillEllipse(RolosColor, (int)((ri_x) - (int)(27 / ControlCnc.Zoom)), (int)ri_y, (int)(54 /
32        ControlCnc.Zoom), (int)(54 / ControlCnc.Zoom));
33    g.DrawEllipse(LinhaBranca, (int)((ri_x) - (int)(27 / ControlCnc.Zoom)), (int)ri_y, (int)(54
34        / ControlCnc.Zoom), (int)(54 / ControlCnc.Zoom));
35
36    //COORDENADAS DO ROLO LATERAL ESQUERDO
37    g.FillEllipse(RolosColor, (int)((ri_x) - (int)(60 / ControlCnc.Zoom)) - (int)(27 /
38        ControlCnc.Zoom), (int)ri_y, (int)(54 / ControlCnc.Zoom), (int)(54 / ControlCnc.Zoom));
39    g.DrawEllipse(LinhaBranca, (int)((ri_x) - (int)(60 / ControlCnc.Zoom)) - (int)(27 /
40        ControlCnc.Zoom), (int)ri_y, (int)(54 / ControlCnc.Zoom), (int)(54 / ControlCnc.Zoom));
41
42    //COORDENADAS DO ROLO LATERAL DIREITO
43    g.FillEllipse(RolosColor, (int)((ri_x) + (int)(60 / ControlCnc.Zoom)) - (int)(27 /
44        ControlCnc.Zoom), (int)ri_y, (int)(54 / ControlCnc.Zoom), (int)(54 / ControlCnc.Zoom));
45    g.DrawEllipse(LinhaBranca, (int)((ri_x) + (int)(60 / ControlCnc.Zoom)) - (int)(27 /
46        ControlCnc.Zoom), (int)ri_y, (int)(54 / ControlCnc.Zoom), (int)(54 / ControlCnc.Zoom));
47
48    // Calculo do ponto extremo
49    // Primeira seccao
50    ArcoCentro ArcoLast = new ArcoCentro();
51
52    if (DadosP.SRaio[1] != 0)
53    {
54        ArcoCentro Arco = new ArcoCentro(370, 420, 0, (int)((DadosP.SRaio[1] / DadosM.Escala) /
55            ControlCnc.Zoom), (int)((DadosP.SComp[1] / DadosM.Escala) / ControlCnc.Zoom));
56        Point[] Pontos2 = new Point[Arco.NGaus];
57        i = 0;
58        for (Cnt = Arco.Start; Cnt <= Arco.End; Cnt++)
59        {
60            Pontos2[i] = new Point((int)Math.Round(Arco.Cx + Arco.RaioT * (Math.Cos(Cnt *
61                (Math.PI / 90F) + (Math.PI / 2)))), (int)Math.Round(Arco.Cy + Arco.RaioT *
62                (Math.Sin(Cnt * (Math.PI / 90F) + (Math.PI / 2)))));
63            i++;
64        }
65        g.DrawLines(LinhaAmarelo, Pontos2);
66        ArcoLast = Arco;
67        Xfinal = ArcoLast.Xfinal;
68        Yfinal = ArcoLast.Yfinal;
69        End = ArcoLast.End;
70    }
71    else
72    {
73        XfinalA = Convert.ToInt32(370 - ((DadosP.SComp[1] / DadosM.Escala) / ControlCnc.Zoom) *
74            (Math.Cos(0 * (Math.PI / 90F))));
75        YfinalA = Convert.ToInt32(420 - ((DadosP.SComp[1] / DadosM.Escala) / ControlCnc.Zoom) *
76            (Math.Sin(0 * (Math.PI / 90F))));
77        g.DrawLine(LinhaAmarelo, 370, 420, XfinalA, YfinalA);
78        Xfinal = XfinalA;
79        Yfinal = YfinalA;
80        End = 0;
81    }
82

```

```

1 // Restantes seccoes
2 if (DadosP.NSeccoes != 1)
3 {
4     bool Active = false;
5     for (int Id = 2; Id <= DadosP.NSeccoes; Id++)
6     {
7         if (DadosP.SRaio[Id] != 0) // seccao circular
8         {
9             ArcoCentro ArcoT = new ArcoCentro(Xfinal, Yfinal, (End * 2),
10                (int)((DadosP.SRaio[Id] / DadosM.Escala) / ControlCnc.Zoom),
11                (int)((DadosP.SComp[Id] / DadosM.Escala) / ControlCnc.Zoom));
12             Point[] PontosT = new Point[ArcoT.NGaus];
13             i = 0;
14             for (Cnt = ArcoT.Start; Cnt <= ArcoT.End; Cnt++)
15             {
16                 PontosT[i] = new Point((int)Math.Round(ArcoT.Cx + ArcoT.RaioT *
17                    (Math.Cos(Cnt * (Math.PI / 90F) + (Math.PI / 2)))), (int)Math.Round(ArcoT.Cy
18                    + ArcoT.RaioT * (Math.Sin(Cnt * (Math.PI / 90F) + (Math.PI / 2)))));
19                 i++;
20             }
21             if (Active == false)
22             {
23                 g.DrawLines(LinhaRed, PontosT);
24                 Active = true;
25             }
26             else
27             {
28                 g.DrawLines(LinhaAmarelo, PontosT);
29                 Active = false;
30             }
31             ArcoLast = ArcoT;
32             Xfinal = ArcoLast.Xfinal;
33             Yfinal = ArcoLast.Yfinal;
34             End = ArcoLast.End;
35         }
36         else // Caso seccao Recta
37         {
38             XfinalA = Convert.ToInt32(ArcoLast.Xfinal - ((DadosP.SComp[Id] / DadosM.Escala)
39                / ControlCnc.Zoom) * (Math.Cos(End * (Math.PI / 90F))));
40             YfinalA = Convert.ToInt32(ArcoLast.Yfinal - ((DadosP.SComp[Id] / DadosM.Escala)
41                / ControlCnc.Zoom) * (Math.Sin(End * (Math.PI / 90F))));
42
43             if (Active == false)
44             {
45                 g.DrawLine(LinhaRed, Xfinal, Yfinal, XfinalA, YfinalA);
46                 Active = true;
47             }
48             else
49             {
50                 g.DrawLine(LinhaAmarelo, Xfinal, Yfinal, XfinalA, YfinalA);
51                 Active = false;
52             }
53             Xfinal = XfinalA;
54             Yfinal = YfinalA;
55         }
56     }
57 }
58 }

```

**Código 15:** Função *private void SimularProgramaDados()*

da primeira secção e com o auxilio da estrutura **struct** **ArcoCentro** para calculo do conjunto de pontos projecta graficamente as seguintes secções, os dados das secções estão armazenados na estrutura do programa **public struct ProgramaDados**.

### **private void DisplayFiles(string dirName)**

O programa *CNC\_CE.exe* foi construído de forma a permitir guardar os programas de regime automático. Os programas são gravados num cartão de memoria *compac flash* de 1GB, nesta memoria está também alojado o sistema operativo *Ms Windows CE* e a aplicação principal *CNC\_CE.exe*. A função *private void DisplayFiles(string dirName)* é usada para mostrar todos os programas guardados na maquina. Corre a directoria referen-

```

1 private void DisplayFiles(string dirName)
2 {
3     string nome;
4     string []campos=new string[2];
5     char[] separador = { ' .' };
6     try
7     {
8         listView1.Items.Clear();
9         // teste para verificar se existe o directorio
10        DirectoryInfo dir = new DirectoryInfo(dirName);
11        if (!dir.Exists)
12            throw new DirectoryNotFoundException("directory does not exist:" + dirName);
13
14        // listagem de directorios
15        foreach (DirectoryInfo di in dir.GetDirectories())
16        {
17            ListViewItem lvi1 = new ListViewItem(new string[] { di.Name.ToString() });
18            lvi1.ImageIndex = 1;
19            listView1.Items.Add(lvi1);
20        }
21        // listagem de ficheiros
22        foreach (FileInfo fi in dir.GetFiles())
23        {
24            nome=fi.Name;
25            campos=nome.Split(separador);
26            ListViewItem lvi1 = new ListViewItem(new string[] { campos[0].ToString() });
27            lvi1.ImageIndex = 1;
28            listView1.Items.Add(lvi1);
29        }
30    }
31    catch (Exception ex)
32    {
33        ex.ToString();
34    }
35 }
36 }

```

**Código 16:** Função *private void DisplayFiles(string dirName)*

ciada e envia para uma *listview* todos os ficheiros encontrados com formato de programa.

### **private void ActualizarEncoder()**

Esta função é responsável pela actualização dos valores dos eixos de controlo da maquina, recebe os valores da comunicação série entre o PLC e o computador industrial num vector de 16bits *Valencoder[]* e envia os valores para a estrutura *public struct PosicaoRolos* que foi inicializada em *PRolos*. É também enviado os valores dos eixos de controlo para os *labels* específico no programa de forma que o operador da maquina tenha conhecimento do valor actual do posicionamento de cada rolo da maquina.



```

1 private void ActualizarEncoder()
2 {
3     this.labPos1E.Text = (ValEncoder[0] / 10.0f).ToString("N", nfi);
4     this.labPos2E.Text = (ValEncoder[1] / 10.0f).ToString("N", nfi);
5     this.labPos3D.Text = (ValEncoder[2] / 10.0f).ToString("N", nfi);
6     this.labPos4D.Text = (ValEncoder[3] / 10.0f).ToString("N", nfi);
7     this.labPos5.Text = (ValEncoder[4] / 10.0f).ToString("N", nfi);
8     this.labPos6.Text = (ValEncoder[5] / 10.0f).ToString("N", nfi);
9     this.labPosRot.Text = ((ValEncoder[6] * DadosM.RelacaoRotacao) / 1.0f).ToString("N", nfi);
10    this.labRotTotal.Text = ((ValEncoder[12] * DadosM.RelacaoRotacao) / 1.0f).ToString("N", nfi);
11    this.labPosL1.Text = (ValEncoder[11] / 10.0f).ToString("N", nfi);
12    this.labPosS.Text = (ValEncoder[7] / 10.0f).ToString("N", nfi);
13    this.labPosL2.Text = (ValEncoder[10] / 10.0f).ToString("N", nfi);
14
15    // VALORES PARA O TEACH IN
16    PRolos.Posicao1E = (ValEncoder[0]);
17    PRolos.Posicao2E = (ValEncoder[1]);
18    PRolos.Posicao3D = (ValEncoder[2]);
19    PRolos.Posicao4D = (ValEncoder[3]);
20    PRolos.Posicao5I = (ValEncoder[4]);
21    PRolos.Posicao6I = (ValEncoder[5]);
22    PRolos.Rotacao = (int)(ValEncoder[6] * DadosM.RelacaoRotacao * 10);
23    PRolos.PosicaoSD = (ValEncoder[11]);
24    PRolos.PosicaoSE = (ValEncoder[10]);
25    PRolos.PosicaoSV = (ValEncoder[7]);
26
27    // VALOR DO ANGULO
28    PRolos.AngRE = (float)(Math.Asin(((PRolos.Posicao1E / 10.0F) - (PRolos.Posicao2E / 10.0F)) /
29        (DadosM.Largura * 10))) * 1800F;
30    PRolos.AngRD = (float)(Math.Asin(((PRolos.Posicao3D / 10.0F) - (PRolos.Posicao4D / 10.0F)) /
31        (DadosM.Largura * 10))) * 1800F;
32    PRolos.AngRI = (float)(Math.Asin(((PRolos.Posicao5I / 10.0F) - (PRolos.Posicao6I / 10.0F)) /
33        (DadosM.Largura * 10))) * 1800F;
34    this.labPosAngD.Text = PRolos.AngRD.ToString("N", nfi) + " °";
35    this.labPosAngE.Text = PRolos.AngRE.ToString("N", nfi) + " °";
36    this.labPosAngI.Text = PRolos.AngRI.ToString("N", nfi) + " °";
37
38
39    if ((BTN_ON == false) || (EstadoMaq.Automatico == true))
40    {
41        SimulacaoRolos();
42    }
43 }

```

**Código 17:** Função *private void ActualizarEncoder()*

**private void Controlo\_Manual()**

```

1 private void Controlo_Manual()
2 {
3     this.labMsnPrincipal.Text = "";
4
5     if ((ValEncoder[8] & 0x0010) == 0x0010)
6     {
7         this.labMsnPrincipal.Text = MSN.SOBE_ROLO_E;
8     }
9     else if ((ValEncoder[8] & 0x0020) == 0x0020)
10    {
11        this.labMsnPrincipal.Text = MSN.DESCE_ROLO_E;
12    }
13    else if ((ValEncoder[8] & 0x0040) == 0x0040)
14    {
15        this.labMsnPrincipal.Text = MSN.SOBE_ROLO_D;
16    }
17    else if ((ValEncoder[8] & 0x0080) == 0x0080)
18    {
19        this.labMsnPrincipal.Text = MSN.DESCE_ROLO_D;
20    }
21    else if ((ValEncoder[8] & 0x0100) == 0x0100)
22    {
23        this.labMsnPrincipal.Text = MSN.SOBE_ROLO_I;
24    }
25    else if ((ValEncoder[8] & 0x0200) == 0x0200)
26    {
27        this.labMsnPrincipal.Text = MSN.DESCE_ROLO_I;
28    }
29    else if ((ValEncoder[8] & 0x0400) == 0x0400)
30    {
31        this.labMsnPrincipal.Text = MSN.RODAR_D;
32    }
33    else if ((ValEncoder[8] & 0x0800) == 0x0800)
34    {
35        this.labMsnPrincipal.Text = MSN.RODAR_E;
36    }
37    else if ((ValEncoder[8] & 0x1000) == 0x1000)
38    {
39        this.labMsnPrincipal.Text = MSN.ABRIR_BRAÇO;
40    }
41    else if ((ValEncoder[8] & 0x2000) == 0x2000)
42    {
43        this.labMsnPrincipal.Text = MSN.FECHAR_BRAÇO;
44    }
45    else if ((ValEncoder[9] & 0x0001) == 0x0001)
46    {
47        this.labMsnPrincipal.Text = MSN.SOBE_ROLO_L1;
48    }
49    else if ((ValEncoder[9] & 0x0002) == 0x0002)
50    {
51        this.labMsnPrincipal.Text = MSN.DESCE_ROLO_L1;
52    }
53    else if ((ValEncoder[9] & 0x0004) == 0x0004)
54    {
55        this.labMsnPrincipal.Text = MSN.DESCE_ROLO_L2;
56    }
57    else if ((ValEncoder[9] & 0x0008) == 0x0008)
58    {
59        this.labMsnPrincipal.Text = MSN.SOBE_ROLO_L2;
60    }
61 }

```

**Código 18:** Função *private void Controlo\_Manual()*

O sistema de comando é composto por dois modos de trabalho, um modo automático em que a máquina executa todos os movimentos dos rolos de forma a cumprir o programa inserido, outro modo de trabalho é o modo manual. No modo manual a máquina está operando de uma forma directa aos comandos exercidos pelo operador.

A função de controlo manual é construída em linguagem de contactos e está alojada no PLC Master, o modo de execução automático ou manual é controlado por uma variável de controlo que impossibilita a máquina operar em modo manual e automático em simultâneo.

A função *private void Controlo\_Manual()* é usada para suporte de comunicação das mensagens emitidas na movimentação dos vários comandos manuais.

A identificação do movimento é pela comparação do conjunto de bits das saídas dos PLC's. As saídas dos PLC's são colocadas em dois registos de 16 bits existentes no PLC Master aos quais o PC tem acesso, é feita a comunicação do PLC Master para o PC onde são guardados em duas variáveis de 16 bits a *ValEncoder[8]* e *ValEncoder[9]*.

A função *private void Controlo\_Manual()* tem a função de comparar o conteúdo das variáveis de registo *ValEncoder[8]* e *ValEncoder[9]* com os registos dos movimentos e enviar a mensagem correspondente para a *label labMsnPrincipal.Text*, esta *label* está visível no centro inferior do programa *CNC\_CE.exe*.

### **private void AUTOMATICO()**

```
1 private void AUTOMATICO()
2 {
3     Ler_Valor_A_POS();
4     switch (VARCONTAUTO)
5     {
6         #region CASE 0 [ FASE LANCAMENTO ACTUALIZAR VARIABEIS DE CONTROLO ]
7         #region CASE 1 [ FASE INICIAL FECHAR BRAÇO, POSICAOES INICIAIS ]
8         #region CASE 2 [ APERTAR CHAPA, RODAR PARA INICIO ]
9         #region CASE 3 [ FASE A DA CALANDRAGEM ]
10        #region CASE 4 [ FASE B DA CALANDRAGEM ]
11        #region CASE 5 [ FASE ACTUALIZAR PASSAGENS ]
12        #region CASE 6 [ FIM DE CALANDRAGEM ]
13        #region CASE 7 [ FASE TEACH IN ]
14        #region CASE 8 [PARALELISMO]
15        #region CASE 9 [FACTOR DE CORRECAO]
16    }
17    #region MENSAGEM PRINCIPAL SELECIONADA ACTIVA
18 }
19 }
```

**Código 19:** Função *private void AUTOMATICO()*

A função *private void AUTOMATICO()* é uma das principais funções desenvolvidas do programa *CNC\_CE.exe*. Esta função gere todo o funcionamento do modo automático, inicializando-se no posicionamento inicial dos rolos, seguindo-se todas as fases de calandragem. Esta função executa os dois modos de trabalho automático o *Teach IN* e o modo automático directo.

A função está estruturada em varias partes que correspondem a fases de calandragem, para melhor executar a tarefa do modo automático foi usado uma estrutura de decisão do tipo *switch* onde implementamos dez expressões correspondendo a dez fases dentro do modo automático.

Vamos apresentar de seguida as dez fases correspondendo ao valor *case* que poderá variar de zero a nove dentro do modo automático.

## CASE 0 [ FASE LANÇAMENTO ACTUALIZAR VARIÁVEIS DE CONTROLO ]

```

1  case 0:
2  if (((ValEncoder[8] & 0x0002) == 0x0002) && (EstadoMaq.TeachIn == false))
3  {
4      VARCONTAUTO = 1;
5      ControlCnc.NumSecActual = 1;
6      ControlCnc.NumPassagem = 1;
7      this.panel6.Visible = true;
8      ControlCnc.ACTIVO = true;
9      Reset_CONTROLROLO();
10     CORRECAO = false;
11     NumCom = 0;
12     FX3U = 5;
13     WRITEFX3U = true;
14     DadosP.NPassagens = NPassagemInicial;
15     DadosP.FCorrecao = FatorCorrecaoInicial;
16     DadosP.FCorrecaoAnt = 0;
17     this.labFactorCorrecao.Text = DadosP.FCorrecao.ToString();
18 }
19 else if (((ValEncoder[8] & 0x0002) == 0x0002) && (EstadoMaq.TeachIn == true))
20 {
21     VARCONTAUTO = 7;
22     ControlCnc.NumPTeachIn = 0;
23     ControlCnc.ACTIVO = true;
24     Reset_CONTROLROLO();
25     NumCom = 0;
26 }
27 else if ((ValEncoder[8] & 0x0002) != 0x0002)
28 {
29     VARCONTAUTO = 0;
30     labMsnPrincipal.Text = MSN.START;// "PRESS [START] TO BEGIN";
31 }
32 break;

```

**Código 20:** Função *CASE 0* - [Fase Inicial]

A fase *CASE = 0* é o início do modo automático, nesta fase a máquina está em espera até que o operador pressione no botão *start* que se encontra no painel de controlo manual, assim que se pressione no botão *start* a condição  $((\text{ValEncoder}[8] \& 0x0002) == 0x0002)$  é verdadeira o processo irá passar para uma nova fase do modo automático. Em função do valor da condição *EstadoMaq.TeachIn == false* irá para a fase um ou para a fase sete. A variável *VARCONTAUTO* é a variável de selecção da estrutura *switch*.

A variável *EstadoMaq.TeachIn* é a variável de controlo que define o modo de automático, se *EstadoMaq.TeachIn == false* a máquina está em modo automático de programação directa, se *EstadoMaq.TeachIn == true* a máquina encontra-se em modo *Teach In* e irá executar um programa criado em modo *Teach In*.

## CASE 1 [ FASE INICIAL FECHAR BRAÇO, POSICOES INICIAIS ]

```

1  case 1:
2  if (ControlCnc.ACTIVO == true)
3  {
4      if (NumCom == 0)
5      {
6          PRolos.OffsetRot = PRolos.Rotacao;
7          FX3U = 1;
8          Address = "M0026";
9          WRITEFX3U = true;
10         NumCom = 1;
11     }
12     else if (NumCom == 1)
13     {
14         PRolos.OffsetRot = 0;
15         CalculoDistancia();
16         RegRolos.Rolo5I = (double)Math.Round((DadosP.Espessura + 10), 1);
17         RegRolos.Rolo6I = (double)Math.Round((DadosP.Espessura + 10), 1);
18         RegRolos.Rolo3D = (double)Math.Round((EstadoMaq.PosInicialChapaInferior +
19             (float)(10 / Math.Cos(DadosM.AngRadMaq))), 1);
20         RegRolos.Rolo4D = (double)Math.Round((EstadoMaq.PosInicialChapaInferior +
21             (float)(10 / Math.Cos(DadosM.AngRadMaq))), 1);
22         RegRolos.Rolo1E = (double)Math.Round((EstadoMaq.PosInicialChapaBatente +
23             (float)(10 / Math.Cos(DadosM.AngRadMaq))), 1);
24         RegRolos.Rolo2E = (double)Math.Round((EstadoMaq.PosInicialChapaBatente +
25             (float)(10 / Math.Cos(DadosM.AngRadMaq))), 1);
26         FX3U = 4;
27         WRITEFX3U = true;
28         NumCom = 2;
29         COM_OK = 0;
30     }
31     else if (NumCom == 2)
32     {
33         if ((ValEncoder[8] & 0x4000) == 0x4000)    // verificar se o braço esta fechado
34         {
35             FX3U = 1;
36             Address = "M0222";
37             WRITEFX3U = true;
38             COM_OK = 0;
39             CONTROL.BRACO = true;
40             ControlCnc.ACTIVO = false;
41             CONTROL.A_POS_INF = true;
42         }
43         else
44         {
45             FX3U = 1;
46             Address = "M0220";
47             WRITEFX3U = true;
48             CONTROL.BRACO = false;
49             ControlCnc.ACTIVO = false;
50             COM_OK = 0;
51             CONTROL.A_POS_FB = true;
52         }
53     }
54 }
55 else
56 {
57     if (CONTROL.BRACO == false)
58     {
59         labMsnPrincipal.Text = MSN.FECHAR_BRAÇO;
60         if (CONTROL.A_POS_FB == false)
61         {
62             FX3U = 1;
63             Address = "M0222";
64             WRITEFX3U = true;
65             CONTROL.A_POS_INF = true;
66             CONTROL.BRACO = true;
67         }
68     }
69     else if ((CONTROL.BRACO == true) && (CONTROL.ROLO_INF == false))
70     {
71         labMsnPrincipal.Text = MSN.POS_ROLO_I;
72         if (CONTROL.A_POS_INF == false)
73         {
74             FX3U = 1;
75             Address = "M0224";
76             WRITEFX3U = true;
77             CONTROL.ROLO_INF = true;
78             CONTROL.A_POS_D = true;
79         }
80     }
81     else if ((CONTROL.ROLO_INF == true) && (CONTROL.ROLO_D == false))
82     {
83         labMsnPrincipal.Text = MSN.POS_ROLO_D;
84         if (CONTROL.A_POS_D == false)
85         {
86             FX3U = 1;
87             Address = "M0223";
88             WRITEFX3U = true;
89             CONTROL.ROLO_D = true;
90             CONTROL.A_POS_E = true;
91         }
92     }
93 }

```

```

1  else if ((CONTROL.ROLO_INF == true) && (CONTROL.ROLO_D == true) && (CONTROL.ROLO_E == false))
2  {
3      labMsnPrincipal.Text = MSN.POS_ROLO_E;
4      if (CONTROL.A_POS_E == false)
5      {
6          CONTROL.ROLO_E = true;
7      }
8  }
9  else if ((CONTROL.ROLO_INF == true) && (CONTROL.ROLO_E == true) && (CONTROL.ROLO_D == true) &&
10  (((ValEncoder[8] & 0x0002) != 0x0002)))
11  {
12      labMsnPrincipal.Text = MSN.CONTINUAR;// "PRESS [START] TO CONTINUE SECOND POSITION";
13  }
14  else if ((CONTROL.ROLO_INF == true) && (CONTROL.ROLO_E == true) && (CONTROL.ROLO_D == true) &&
15  (((ValEncoder[8] & 0x0002) == 0x0002)))
16  {
17      labMsnPrincipal.Text = "";
18      VARCONTAUTO = 2;
19      ControlCnc.ACTIVO = true;
20      NumCom = 0;
21      Reset_CONTROLROLO();
22  }
23  }
24  break;

```

**Código 21:** Função *CASE 1* - [Fase 1ª Posicionamento]

Na fase um é dado o início para a movimentação dos rolos em modo automático. A primeira fase de posicionamento dos rolos da máquina consiste em colocar os rolos de forma a introduzir a chapa na máquina podemos observar na Figura 3.4 Fase 1.

Esta parte de código da função *public void AUTOMATICO()* está dividida em partes pela variável *NumCom*. Na primeira parte é feita a anulação do registo de rotação, passando de seguida para o cálculo da posição de cada rolo da máquina em função da espessura da chapa e calculado também o comprimento de chapa que está entre o rolo inferior e o rolo lateral. Nas fases seguintes é feito o posicionamento de cada rolo em função dos valores calculados, e feita uma comunicação ao PLC Master onde é colocado os valores do posicionamento final de cada rolo é dado o início do posicionamento do rolo inferior, após terminar o posicionamento do rolo inferior é inicializado o posicionamento do rolo lateral direito e em seguida o posicionamento do rolo lateral esquerdo.

O programa no fim do posicionamento do rolo lateral esquerdo fica a aguardar um novo sinal do botão *start* para passar para a fase dois.

## CASE 2 [ APERTAR CHAPA]

Na segunda fase do processo de calandragem em modo automático é enviado para o PLC Master os novos valores dos registos de posicionamento dos rolos, nesta fase difere da anterior o rolo inferior que passa para um posicionamento igual á espessura programada e o rolo lateral esquerdo irá baixar para o mesmo valor do rolo lateral direito.

Após o envio dos valores de posicionamento para o PLC Master é enviado também um bit que irá activar o posicionamento do rolo inferior, a função de posicionamento do rolo inferior no PLC Master irá retornar um bit activo quando finalizar o posicionamento, este

```

1  case 2:
2  if (ControlCnc.ACTIVO == true)
3  {
4      if (NumCom == 0)
5      {
6          PRolos.OffsetRot = PRolos.Rotacao;
7          FX3U = 1;
8          Address = "M0026";
9          WRITEFX3U = true;
10         NumCom = 1;
11         PosOposto1 = 0;
12         PosOposto = (double)Math.Round(EstadoMaq.PosInicialChapaInferior +
13             (float)(DadosP.Espessura / Math.Cos(DadosM.AngRadMaq)) + 7.5f, 1);
14     }
15     else if (NumCom == 1)
16     {
17         RegRolos.Rolo5I = (double)Math.Round((DadosP.Espessura),1);
18         RegRolos.Rolo6I = (double)Math.Round((DadosP.Espessura),1);
19         RegRolos.Rolo1E = PosOposto;
20         RegRolos.Rolo2E = PosOposto;
21         RegRolos.Rolo3D = PosOposto;
22         RegRolos.Rolo4D = PosOposto;
23         FX3U = 4;
24         WRITEFX3U = true;
25         NumCom = 2;
26     }
27     else if (NumCom == 2)
28     {
29         FX3U = 1;
30         Address = "M0222";
31         WRITEFX3U = true;
32         ControlCnc.ACTIVO = false;
33         CONTROL.A_POS_INF = true;
34     }
35 }
36 else
37 {
38     if (CONTROL.ROLO_INF == false)
39     {
40         labMsnPrincipal.Text = MSN.POS_ROLO_I;
41         if (CONTROL.A_POS_INF == false)
42         {
43             FX3U = 1;
44             Address = "M0224";
45             WRITEFX3U = true;
46             CONTROL.ROLO_INF = true;
47             CONTROL.A_POS_D = true;
48         }
49     }
50     else if ((CONTROL.ROLO_INF == true) && (CONTROL.ROLO_D == false))
51     {
52         labMsnPrincipal.Text = MSN.POS_ROLO_D;
53         if (CONTROL.A_POS_D == false)
54         {
55             CONTROL.ROLO_D = true;
56         }
57     }
58     else if ((CONTROL.ROLO_INF == true) && (CONTROL.ROLO_D == true) &&
59         (((ValEncoder[8] & 0x0002) != 0x0002) && (this.EstadoMaq.Avanco != true)))
60     {
61         labMsnPrincipal.Text = MSN.CONTINUAR;// "PRESS [START] TO CONTINUE THIRD POSITION";
62     }
63     else if ((CONTROL.ROLO_INF == true) && (CONTROL.ROLO_D == true) &&
64         (((ValEncoder[8] & 0x0002) == 0x0002) && (this.EstadoMaq.Avanco != true)))
65     {
66         labMsnPrincipal.Text = "";
67         VARCONTAUTO = 3;
68         ControlCnc.ACTIVO = true;
69         NumCom=0;
70         Reset_CONTROLROLO();
71     }
72     else if ((CONTROL.ROLO_INF == true) && (CONTROL.ROLO_D == true) &&
73         (this.EstadoMaq.Avanco == true))
74     {
75         labMsnPrincipal.Text = "";
76         VARCONTAUTO = 3;
77         ControlCnc.ACTIVO = true;
78         NumCom=0;
79         Reset_CONTROLROLO();
80     }
81 }
82 break;

```

Código 22: Função CASE 2 - [Fase aperto da chapa]

bit de retorno irá dar início ao envio de um novo bit para posicionamento do rolo lateral esquerdo, deste modo se processa a comunicação entre o PC e o PLC Master.

### CASE 3 e 4 [ FASE A e B DA CALANDRAGEM ]

```

1  case 3:
2  if (CORRECAO == false)
3  {
4      auxiliar = DadosP.NPassagens - ControlCnc.NumPassagem + 1;
5  }
6  else
7  {
8      auxiliar = 1;
9  }
10 SENTIDO_A = ControlCnc.NumPassagem % 2;
11 if (SENTIDO_A == 1)
12 {
13     FASE_1_SENTIDO_B_NEW(DadosP.SRaio[ControlCnc.NumSecActual] * auxiliar);
14 }
15 else
16 {
17     FASE_1_SENTIDO_A_NEW(DadosP.SRaio[ControlCnc.NumSecActual] * auxiliar);
18 }
19 }
20 break;
21
22
23
24
25
26
27 case 4:
28 if (CORRECAO == false)
29 {
30     auxiliar = DadosP.NPassagens - ControlCnc.NumPassagem + 1;
31 }
32 else
33 {
34     auxiliar = 1;
35 }
36 }
37 SENTIDO_A = ControlCnc.NumPassagem % 2;
38 if (SENTIDO_A == 1)
39 {
40     FASE_2_SENTIDO_B(DadosP.SRaio[ControlCnc.NumSecActual] * auxiliar);
41 }
42 else
43 {
44     FASE_2_SENTIDO_A(DadosP.SRaio[ControlCnc.NumSecActual] * auxiliar);
45 }
46 }
47 break;
48

```

**Código 23:** Função *CASE 3* - [Fase A e B da Calandragem]

Após o início da calandragem na fase anterior é iniciado o processo de calandragem referente as fases 3,4,5,6,7 e 8 da Figura 3.4, na primeira fase é realizado as fases 3,4,5 e 6 onde é feita a calandragem numa das extremidades da secção esta tarefa é realizada pela função *FASE\_1\_SENTIDO\_A\_NEW()* ou *FASE\_1\_SENTIDO\_B\_NEW()* são duas funções semelhantes que diferem no sentido de execução uma com sentido directo outra para o sentido inverso, após finalizar a fase de calandragem 6 da Figura 3.4 é iniciada a segunda parte da calandragem, esta é executada pela função *FASE\_2\_SENTIDO\_A()* ou *FASE\_2\_SENTIDO\_B()* que são semelhantes, diferem apenas no sentido de rotação, uma está construída para o sentido directo e outra para o sentido inverso da rotação.



## CASE 5 [ FASE ACTUALIZAR PASSAGENS ]

```

1  case 5:
2  SENTIDO_A = ControlCnc.NumPassagem % 2;
3  if (SENTIDO_A == 1)
4  {
5      ControlCnc.NumSecActual++;
6  }
7  else
8  {
9      ControlCnc.NumSecActual--;
10 }
11 if ((ControlCnc.NumSecActual > this.DadosP.NSeccoes) || (ControlCnc.NumSecActual <= 0))
12 {
13     ControlCnc.NumPassagem++;
14     if (ControlCnc.NumPassagem > this.DadosP.NPassagens)
15     {
16         VARCONTAUTO = 9;
17         ControlCnc.ACTIVO = true;
18         CORRECAO = false;
19         DadosP.FCorrecaoAnt = DadosP.FCorrecao;
20         PRolos.OffsetRot = 0;
21         btnMaisFactor.Enabled = true;
22         btnMenosFactor.Enabled = true;
23         this.btnActivar.Enabled = true;
24         this.ACT_CORRECAO = false;
25     }
26     else
27     {
28         if (ControlCnc.NumSecActual == (this.DadosP.NSeccoes + 1))
29         {
30             ControlCnc.NumSecActual = this.DadosP.NSeccoes;
31             VARCONTAUTO = 3;
32             ControlCnc.ACTIVO = true;
33             PRolos.OffsetRot = 0;
34         }
35         else
36         {
37             ControlCnc.NumSecActual = 1;
38             VARCONTAUTO = 3;
39             ControlCnc.ACTIVO = true;
40             PRolos.OffsetRot = 0;
41             PRolos.Rotacao = 0;
42         }
43     }
44 }
45 else
46 {
47     VARCONTAUTO = 3;
48     ControlCnc.ACTIVO = true;
49 }
50 break;

```

Código 24: Função CASE 5 - [Fase Actualização Passagens]

Normalmente um programa é composto por mais que uma passagem ou secções, esta fase encontra-se após terminarmos um processo de calandragem de uma secção, aqui irá ser incrementado o numero da secção actual e verificado se foi atingido o limite de secções programadas, se o limite de secções foi atingido será verificado o numero de passagens, se o numero de passagens total não foi atingido será dado inicio a uma nova passagem no sentido inverso, este processo irá decorrer até executar todas as secções e passagens programadas. Quando for verificado o final da ultima secção da ultima passagem o programa ira passar para a CASE 9 [FACTOR DE CORRECAO] caso não esteja nesta situação ira passar para o inicio do ciclo de calandragem CASE 3 e 4 [FASE A e B DA CALANDRAGEM].

## CASE 6 [ FIM DE CALANDRAGEM ]

```

1 case 6:
2 if (ControlCnc.ACTIVO == true)
3 {
4     if (EstadoMaq.AbreBraco == true)
5     {
6         if (NumCom == 0)
7         {
8             CalculoDistancia();
9             RegRolos.Rolo5I = (double)Math.Round((DadosM.CursorInferior * 0.65F),1);
10            RegRolos.Rolo6I = (double)Math.Round((DadosM.CursorInferior * 0.65F),1);
11            RegRolos.Rolo1E = (double)Math.Round((EstadoMaq.PosInicialChapaInferior +
12                (float)(DadosP.Espessura / Math.Cos(DadosM.AngRadMaq)) + 10),1);
13            RegRolos.Rolo2E = (double)Math.Round((EstadoMaq.PosInicialChapaInferior +
14                (float)(DadosP.Espessura / Math.Cos(DadosM.AngRadMaq)) + 10),1);
15            RegRolos.Rolo3D = (double)Math.Round((EstadoMaq.PosInicialChapaInferior +
16                (float)(DadosP.Espessura / Math.Cos(DadosM.AngRadMaq)) + 10),1);
17            RegRolos.Rolo4D = (double)Math.Round((EstadoMaq.PosInicialChapaInferior +
18                (float)(DadosP.Espessura / Math.Cos(DadosM.AngRadMaq)) + 10),1);
19            FX3U = 4;
20            WRITEFX3U = true;
21            NumCom = 1;
22        }
23        else if (NumCom == 1)
24        {
25            FX3U = 1;
26            Address = "M0222";
27            WRITEFX3U = true;
28            this.btnMaisFactor.Enabled = false;
29            this.btnMenosFactor.Enabled = false;
30            CONTROL.A_POS_INF = true;
31            CONTROL.BRACO = false;
32            NumCom = 0;
33            CORRECAO = false;
34            ControlCnc.ACTIVO = false;
35        }
36    }
37    else
38    {
39        this.labMsnPrincipal.Text = "";
40        VARCONTAUTO = 0;
41        ControlCnc.ACTIVO = true;
42        CORRECAO = false;
43        this.btnMaisFactor.Enabled = false;
44        this.btnMenosFactor.Enabled = false;
45    }
46 }
47 else
48 {
49     if (CONTROL.ROLO_INF == false)
50     {
51         this.labMsnPrincipal.Text = MSN.POS_ROLO_I;
52         if (CONTROL.A_POS_INF == false)
53         {
54             FX3U = 1;
55             Address = "M0223";
56             WRITEFX3U = true;
57             CONTROL.ROLO_INF = true;
58             CONTROL.A_POS_E = true;
59         }
60     }
61     else if ((CONTROL.ROLO_INF == true) && (CONTROL.ROLO_E == false))
62     {
63         this.labMsnPrincipal.Text = MSN.POS_ROLO_I;
64         if (CONTROL.A_POS_E == false)
65         {
66             FX3U = 1;
67             Address = "M0224";
68             WRITEFX3U = true;
69             CONTROL.ROLO_E = true;
70             CONTROL.A_POS_D = true;
71         }
72     }
73     else if ((CONTROL.ROLO_INF == true) && (CONTROL.ROLO_E == true) && (CONTROL.ROLO_D == false))
74     {
75         this.labMsnPrincipal.Text = MSN.POS_ROLO_D;
76         if (CONTROL.A_POS_D == false)
77         {
78             FX3U = 1;
79             Address = "M0221";
80             WRITEFX3U = true;
81             CONTROL.ROLO_D = true;
82             CONTROL.A_POS_AB = true;
83         }
84     }
85     else if ((CONTROL.ROLO_INF == true) && (CONTROL.ROLO_E == true) && (CONTROL.ROLO_D == true) &&
86         (CONTROL.BRACO == false))
87     {
88         this.labMsnPrincipal.Text = MSN.ABRIR_BRAÇO;
89         if (CONTROL.A_POS_AB == false)
90         {
91             CONTROL.BRACO = true;
92         }
93     }
94 }

```

```

1     else if ((CONTROL.ROLO_INF == true) && (CONTROL.ROLO_E == true) && (CONTROL.ROLO_D == true) &&
2         (CONTROL.BRACO == true) && (((ValEncoder[8] & 0x0002) != 0x0002)))
3     {
4         this.labMsnPrincipal.Text = "FIM DE PROGRAMA";
5     }
6     else if ((CONTROL.ROLO_INF == true) && (CONTROL.ROLO_E == true) && (CONTROL.ROLO_D == true) &&
7         (CONTROL.BRACO == true) && (((ValEncoder[8] & 0x0002) == 0x0002)))
8     {
9         this.labMsnPrincipal.Text = "";
10        VARCONTAUTO = 0;
11        ControlCnc.ACTIVO = true;
12        NumCom = 0;
13        Reset_CONTROLROLO();
14    }
15 }
16 break;

```

**Código 25:** Função *CASE 6 - [Fim de Calandragem]*

Esta fase está no final do processo de calandragem, após a fase de calandragem e a fase de correcção de calandragem surge o posicionamento final, esta fase consiste em posicionar ou não os rolos em modo automático de forma a retirar a chapa da máquina. Esta fase é controlada por uma variável de controlo que é possível ser activa ou desactiva pelo operador no decorrer do processo de calandragem, a variável `EstadoMaq.AbreBraco` é accionada por um botão que esta presente no *menu* automático.

## CASE 7 [ FASE TEACH IN ]

```

1 case 7:
2 if (ControlCnc.ACTIVO == true)
3 {
4     if (NumCom == 0)
5     {
6         FX3U = 1;
7         Address = "M0026";
8         WRITEFX3U = true;
9         NumCom = 1;
10    }
11    else if (NumCom == 1)
12    {
13        FX3U = 8;
14        WRITEFX3U = true;
15        NumCom = 2;
16    }
17    else if (NumCom == 2)
18    {
19        this.labTeachIn.Text = "TEACH IN [ " + ControlCnc.NumPTeachIn.ToString() + " ]";
20        if ((ValEncoder[8] & 0x4000) == 0x4000) // verificar se o braço esta fechado
21        {
22            FX3U = 1;
23            Address = "M0223";
24            WRITEFX3U = true;
25            CONTROL.BRACO = true;
26            CONTROL.A_POS_E = true;
27            ControlCnc.ACTIVO = false;
28        }
29        else
30        {
31            FX3U = 1;
32            Address = "M0220";
33            WRITEFX3U = true;
34            CONTROL.BRACO = false;
35            CONTROL.A_POS_FB = true;
36            ControlCnc.ACTIVO = false;
37        }
38        NumCom = 0;
39    }
40 }

```

O modo automático *Teach IN* é um modo automático criado com base num processo de calandragem manual, onde é guardado todos os valores de cada posicionamento dos eixos de posicionamento da máquina. Este processo dá origem a um ficheiro que é guardado na máquina e pode ser usado repetidas vezes. Dentro da função *private void AUTOAMTICO()* existe uma parte para executar um programa do tipo *Teach In*, nesta secção de código é enviado o valor para o PLC Master de cada posicionamento referente a passagem actual, em seguida é dado o início do posicionamento de todos os eixos de controlo, no final do posicionamento é iniciada nova sequência, o processo será finalizado assim que forem executadas todas as passagens capturadas em modo manual.

```

1  else
2  {
3      if (CONTROL.BRACO == false)
4      {
5          this.labMsnPrincipal.Text = MSN.FECHAR_BRAÇO;
6          if (CONTROL.A_POS_FB == false)
7          {
8              FX3U = 1;
9              Address = "M0223";
10             WRITEFX3U = true;
11             CONTROL.BRACO = true;
12             CONTROL.A_POS_E = true;
13         }
14     }
15     else if ((CONTROL.BRACO == true) && (CONTROL.ROLO_E == false))
16     {
17         this.labMsnPrincipal.Text = MSN.POS_ROLO_E;
18         if (CONTROL.A_POS_E == false)
19         {
20             FX3U = 1;
21             Address = "M0222";
22             WRITEFX3U = true;
23             CONTROL.ROLO_E = true;
24             CONTROL.A_POS_INF = true;
25         }
26     }
27     else if ((CONTROL.ROLO_E == true) && (CONTROL.ROLO_INF == false))
28     {
29         this.labMsnPrincipal.Text = MSN.POS_ROLO_I;
30         if (CONTROL.A_POS_INF == false)
31         {
32             FX3U = 1;
33             Address = "M0224";
34             WRITEFX3U = true;
35             CONTROL.ROLO_INF = true;
36             CONTROL.A_POS_D = true;
37         }
38     }
39     else if ((CONTROL.ROLO_E == true) && (CONTROL.ROLO_INF == true) && (CONTROL.ROLO_D == false))
40     {
41         this.labMsnPrincipal.Text = MSN.POS_ROLO_D;
42         if (CONTROL.A_POS_D == false)
43         {
44             if (DadosM.SUP1 != 0)
45             {
46                 FX3U = 1;
47                 Address = "M0228";
48                 WRITEFX3U = true;
49                 CONTROL.A_POS_L3 = true;
50             }
51             CONTROL.ROLO_D = true;
52         }
53     }
54     else if ((CONTROL.ROLO_E == true) && (CONTROL.ROLO_INF == true) && (CONTROL.ROLO_D == true) &&
55             (CONTROL.ROLO_L3 == false))
56     {
57         this.labMsnPrincipal.Text = MSN.POS_ROLO_L1;
58         if ((CONTROL.A_POS_L3 == false) || (DadosM.SUP1 == 0))
59         {
60             if (DadosM.SUP3 != 0)
61             {
62                 FX3U = 1;
63                 Address = "M0226";
64                 WRITEFX3U = true;
65                 CONTROL.A_POS_L1 = true;
66             }
67             CONTROL.ROLO_L3 = true;
68         }
69     }
70 }
71 else if ((CONTROL.ROLO_E == true) && (CONTROL.ROLO_INF == true) && (CONTROL.ROLO_D == true) &&
72         (CONTROL.ROLO_L3 == true) && (CONTROL.ROLO_L1 == false))
73 {
74     this.labMsnPrincipal.Text = MSN.POS_ROLO_L2;
75     if ((CONTROL.A_POS_L1 == false) || (DadosM.SUP3 == 0))
76     {
77         if (DadosM.SUP2 != 0)
78         {
79             FX3U = 1;
80             Address = "M0227";
81             WRITEFX3U = true;
82             CONTROL.A_POS_L2 = true;
83         }
84         CONTROL.ROLO_L1 = true;
85     }
86 }
87 else if ((CONTROL.ROLO_E == true) && (CONTROL.ROLO_INF == true) && (CONTROL.ROLO_D == true) &&
88         (CONTROL.ROLO_L1 == true) && (CONTROL.ROLO_L3 == true) && (CONTROL.ROLO_L2 == false))
89 {
90     this.labMsnPrincipal.Text = MSN.POS_ROLO_L3;
91     if ((CONTROL.A_POS_L2 == false) || (DadosM.SUP2 == 0))
92     {
93         FX3U = 1;
94         Address = "M0225";
95         WRITEFX3U = true;
96     }
97 }
98 }
99 }

```

```

1      CONTROL.ROLO_L2 = true;
2      CONTROL.A_POS_ROT = true;
3  }
4  }
5  else if ((CONTROL.ROLO_E == true) && (CONTROL.ROLO_INF == true) && (CONTROL.ROLO_D == true) &&
6          (CONTROL.ROLO_L1 == true) && (CONTROL.ROLO_L3 == true) && (CONTROL.ROLO_L2 == true) &&
7          (CONTROL.ROLO_ROT == false))
8  {
9      if (CONTROL.A_POS_ROT == false)
10     {
11         CONTROL.ROLO_ROT = true;
12     }
13     this.labMsnPrincipal.Text = MSN.POS_ROLO_ROT;
14 }
15 else if ((CONTROL.ROLO_E == true) && (CONTROL.ROLO_INF == true) && (CONTROL.ROLO_D == true) &&
16         (CONTROL.ROLO_L1 == true) && (CONTROL.ROLO_L3 == true) && (CONTROL.ROLO_L2 == true) &&
17         (CONTROL.ROLO_ROT == true) && ((ValEncoder[8] & 0x0002) != 0x0002) &&
18         (this.EstadoMaq.Avanco != true))
19 {
20     this.labMsnPrincipal.Text = MSN.CONTINUAR;
21 }
22 else if ((CONTROL.ROLO_E == true) && (CONTROL.ROLO_INF == true) && (CONTROL.ROLO_D == true) &&
23         (CONTROL.ROLO_L1 == true) && (CONTROL.ROLO_L3 == true) && (CONTROL.ROLO_L2 == true) &&
24         (CONTROL.ROLO_ROT == true) && ((ValEncoder[8] & 0x0002) != 0x0002) &&
25         (ControlCnc.NumPTeachIn == 1))
26 {
27     this.labMsnPrincipal.Text = MSN.CONTINUAR;
28 }
29 else if ((CONTROL.ROLO_E == true) && (CONTROL.ROLO_INF == true) && (CONTROL.ROLO_D == true) &&
30         (CONTROL.ROLO_L1 == true) && (CONTROL.ROLO_L3 == true) && (CONTROL.ROLO_L2 == true) &&
31         (CONTROL.ROLO_ROT == true) && ((ValEncoder[8] & 0x0002) == 0x0002) &&
32         (this.EstadoMaq.Avanco != true))
33 {
34     ControlCnc.NumPTeachIn++;
35     if (ControlCnc.NumPTeachIn <= ControlCnc.NumTPTeachIn)
36     {
37         ControlCnc.ACTIVO = true;
38         Reset_CONTROLROLO();
39     }
40     else
41     {
42         VARCONTAUTO = 6;
43         ControlCnc.ACTIVO = true;
44         Reset_CONTROLROLO();
45     }
46 }
47 else if ((CONTROL.ROLO_E == true) && (CONTROL.ROLO_INF == true) && (CONTROL.ROLO_D == true) &&
48         (CONTROL.ROLO_L1 == true) && (CONTROL.ROLO_L3 == true) && (CONTROL.ROLO_L2 == true) &&
49         (CONTROL.ROLO_ROT == true) && ((ValEncoder[8] & 0x0002) == 0x0002))
50 {
51     ControlCnc.NumPTeachIn++;
52     if (ControlCnc.NumPTeachIn <= ControlCnc.NumTPTeachIn)
53     {
54         ControlCnc.ACTIVO = true;
55         Reset_CONTROLROLO();
56     }
57     else
58     {
59         VARCONTAUTO = 6;
60         ControlCnc.ACTIVO = true;
61         Reset_CONTROLROLO();
62     }
63 }
64 else if ((CONTROL.ROLO_E == true) && (CONTROL.ROLO_INF == true) && (CONTROL.ROLO_D == true) &&
65         (CONTROL.ROLO_L1 == true) && (CONTROL.ROLO_L3 == true) && (CONTROL.ROLO_L2 == true) &&
66         (CONTROL.ROLO_ROT == true) && (this.EstadoMaq.Avanco == true) &&
67         (ControlCnc.NumPTeachIn != 1))
68 {
69     ControlCnc.NumPTeachIn++;
70     if (ControlCnc.NumPTeachIn <= ControlCnc.NumTPTeachIn)
71     {
72         ControlCnc.ACTIVO = true;
73         Reset_CONTROLROLO();
74     }
75     else
76     {
77         VARCONTAUTO = 6;
78         ControlCnc.ACTIVO = true;
79         Reset_CONTROLROLO();
80     }
81 }
82 }
83 break;

```

Código 26: Função CASE 7 - [Fase Teach In]

## CASE 8 [PARALELISMO]

```

1  case 8:
2  if (ControlCnc.ACTIVO == true)
3  {
4      if (NumCom == 0)
5      {
6          FX3U = 1;
7          Address = "M0229";
8          WRITEFX3U = true;
9          NumCom = 1;
10     }
11     else if (NumCom == 1)
12     {
13         FX3U = 1;
14         Address = "M0223";
15         WRITEFX3U = true;
16         CONTROL.A_POS_E = true;
17         ControlCnc.ACTIVO = false;
18     }
19 }
20 else
21 {
22     if (CONTROL.ROLO_E == false)
23     {
24         this.labMsnPrincipal.Text = MSN.NIVELAR;
25         if (CONTROL.A_POS_E == false)
26         {
27             FX3U = 1;
28             Address = "M0222";
29             WRITEFX3U = true;
30             CONTROL.A_POS_INF = true;
31             CONTROL.ROLO_E = true;
32         }
33     }
34     else if ((CONTROL.ROLO_E == true) && (CONTROL.ROLO_INF == false))
35     {
36         this.labMsnPrincipal.Text = MSN.NIVELAR;
37         if (CONTROL.A_POS_INF == false)
38         {
39             FX3U = 1;
40             Address = "M0224";
41             WRITEFX3U = true;
42             CONTROL.A_POS_D = true;
43             CONTROL.ROLO_INF = true;
44         }
45     }
46     else if ((CONTROL.ROLO_E == true) && (CONTROL.ROLO_INF == true) && (CONTROL.ROLO_D == false))
47     {
48         this.labMsnPrincipal.Text = MSN.NIVELAR;
49         if (CONTROL.A_POS_D == false)
50         {
51             FX3U = 1;
52             Address = "M0230";
53             WRITEFX3U = true;
54             CONTROL.ROLO_D = true;
55             EstadoMq.Automatico = false;
56             NumCom = 0;
57         }
58     }
59 }
60 break;

```

Código 27: Função CASE 8 - [Paralelismo]

Função de auxílio ao modo de nivelar a máquina em modo manual, quando se pressiona no botão *start* em modo manual a máquina inicia um nivelamento de todos os rolos, este processo é gerido por esta parte de código presente na secção CASE 8.

## CASE 9 [FACTOR DE CORRECAO]

Está função permite ao operador no final de cada calandragem optar por corrigir a calandragem efectuada pela máquina em modo automático, este processo consiste em adicionar ao valor do posicionamento dos rolos laterais uma quantidade a definir pelo operador

```

1  case 9:
2  if ((ValEncoder[8] & 0x0002) != 0x0002)
3  {
4      this.labMsnPrincipal.Text = "FIM DE PROGRAMA - FACTOR CORREÇÃO";
5  }
6  else if (((ValEncoder[8] & 0x0002) == 0x0002)&&(ACT_CORRECAO == true))
7  {
8      this.ACT_CORRECAO = false;
9      this.btnActivar.Enabled = false;
10     this.labMsnPrincipal.Text = "";
11     ControlCnc.ACTIVO = true;
12     ActivarFactor();
13     if (CORRECAO == true)
14     {
15         this.labMsnPrincipal.Text = "";
16         this.btnMaisFactor.Enabled = false;
17         this.btnMenosFactor.Enabled = false;
18         this.DadosP.NPassagens++;          // INCREMENTAR NUMERO DE PASSAGEM
19         VARCONTAUTO = 5;
20     }
21     else
22     {
23         this.labMsnPrincipal.Text = "";
24         this.btnMaisFactor.Enabled = false;
25         this.btnMenosFactor.Enabled = false;
26         VARCONTAUTO = 6;
27     }
28 }
29 break;

```

**Código 28:** Função *CASE 9 - [Factor de Correção]*

da máquina. Este processo pode ser repetido varias vezes para a mesma peça.

**private float PosicaoRoloLateral(float RaioActual)**

```

1  private float PosicaoRoloLateral(float RaioActual)
2  {
3      double Angulo = (float)(this.DadosM.AnguloCarac * Math.PI) / 180;
4      double AngDes = (float)(this.DadosM.DesvioAngular * Math.PI) / 180;
5      double DifAng = (float)((this.DadosM.AnguloCarac - this.DadosM.DesvioAngular) * Math.PI) / 180;
6      double lado_b, lado_c, lado_d, lado_e, lado_g;
7
8      double PosRaio, PosMin;
9      double vertice_2;
10     double lateral_1, lateral_2;
11     double CentroRaio;
12     double comp_1, comp_2, comp_3;
13
14     // CALCULO DO COMPRIMENTO MINIMO
15     lado_b = this.DadosM.DistrSVertice * Math.Cos(AngDes);
16     lado_c = this.DadosM.DistrSVertice * Math.Sin(AngDes);
17     lado_e = lado_c / (Math.Sin(DifAng));
18     lado_d = lado_e * Math.Cos(DifAng);
19     vertice_2 = lado_d + lado_b;
20     lateral_1 = vertice_2 * Math.Cos(DifAng);
21     lado_g = vertice_2 * Math.Sin(DifAng);
22     lateral_2 = Math.Sqrt((this.DadosM.DistrLatRSup * this.DadosM.DistrLatRSup) - (lado_g * lado_g));
23     PosMin = lateral_2 + lateral_1;
24
25     // CALCULO DO COMPRIMENTO PARA O RAIOS
26     CentroRaio = ((vertice_2 + (this.DadosM.DiamRoloSuperior / 2.0f)) - RaioActual);
27     comp_1 = CentroRaio * Math.Cos(DifAng);
28     comp_2 = CentroRaio * Math.Sin(DifAng);
29     comp_3 = Math.Sqrt(((RaioActual + (this.DadosM.DiamRoloLateral / 2.0f)) * (RaioActual +
30     (this.DadosM.DiamRoloLateral / 2.0f))) - (comp_2 * comp_2));
31     PosRaio = ((comp_1 + comp_3) - (PosMin - this.DadosP.Espessura));
32
33     return (float)PosRaio;
34 }

```

**Código 29:** Função *private float PosicaoRoloLateral(float RaioActual)*

Esta função danos o valor do posicionamento dos rolos laterais em função de um raio pretendido. A função apresentada é a aplicação do algoritmo desenvolvido apresentado no



capítulo 4 designado de Segundo Algoritmo de Posicionamento onde resultou a equação 4.4. É usada sempre que necessitamos de posicionar os rolos laterais para obter uma curvatura desejada.

A função é chamada dentro das funções `FASE_1_SENTIDO_A_NEW()`, `FASE_1_SENTIDO_B_NEW()`, `FASE_2_SENTIDO_A()` e `FASE_2_SENTIDO_B()`.

**private void CalculoDistancia()**

```

1 private void CalculoDistancia()
2 {
3     double angulo1 = (float)(this.DadosM.AnguloCarac* Math.PI) / 180f;
4     this.EstadoMaq.DistRLatEixoInfMin = (((this.DadosM.DistRSVertice +
5     (this.DadosM.DiamRoloSuperior/2.0f) + (this.DadosP.Espessura / 2)) * ((float)Math.Sin(angulo1) /
6     (float)Math.Cos(angulo1))) - (this.DadosM.DiamRoloLateral/2.0f));
7
8     float a = (float)(this.DadosM.DistRSVertice * (Math.Sin(angulo1)));
9     float h = (this.DadosM.DiamRoloLateral / 2.0f) + (this.DadosM.DiamRoloSuperior / 2.0f);
10    float hh = h * h;
11    float aa = a * a;
12    float b = (float)(Math.Sqrt(hh - aa));
13    float c = (float)(this.DadosM.DistRSVertice * (Math.Cos(angulo1)));
14    this.EstadoMaq.DistV_RLatMin = b + c;
15
16    float l = this.DadosM.DistRSVertice + (this.DadosM.DiamRoloSuperior / 2.0f) +
17    (this.DadosP.Espessura / 2.0f);
18    this.EstadoMaq.PosInicialChapaBatente = ((l / (float)Math.Cos(angulo1)) -
19    this.EstadoMaq.DistV_RLatMin);
20
21    this.EstadoMaq.PosInicialChapaInferior = this.EstadoMaq.PosInicialChapaBatente +
22    (this.DadosM.DiamRoloLateral / 2.0f) + (float)((this.DadosP.Espessura / 2.0f) /
23    Math.Cos(angulo1));
24 }

```

**Código 30:** Função *private void CalculoDistancia()*

Função usada para calculo do comprimento de chapa entre o centro rolo inferior e o rolo lateral, e também para calculo da posição do rolo lateral abaixo da linha da chapa na posição inicial.

### 6.2.6 Menus do programa *CNC\_CE.exe*

O programa esta dividido em cinco *menus* como já referimos, passamos a descrever:

## MENU MANUAL

No *menu* manual a maquina esta em modo de funcionamento manual, os comandos dos manípulos accionarão directamente a maquina, toda a parte de controlo de accionamento é executado nos PLCs. No PC é feita a comunicação com o PLC Master onde constantemente se actualiza o posicionamento de todos os rolos, mantendo uma simulação grafia do posicionamento real.

Na Figura 6.9 podemos observar o layout gráfico do *Menu* Manual.

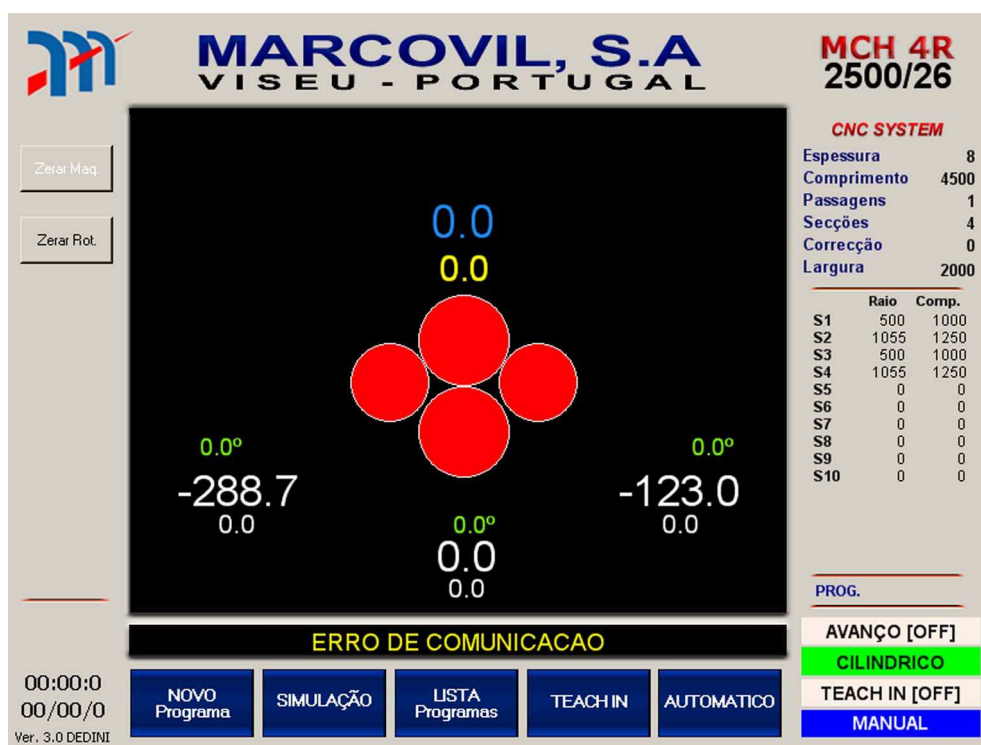


Figura 6.9: Imagem do Programa CNC MODO MANUAL

## NOVO PROGRAMA

Aqui o operador têm a possibilidade de construir um programa de conformação de chapa, basta introduzir nos campos ESPESSURA, COMPRIMENTO, PASSAGENS, SECÇÕES, CORRECÇÃO e LARGURA, para se ter um programa para execução do controlo numérico computarizado.

Na Figura 6.10 podemos observar o layout gráfico do *Menu* Novo programa.

## MENU SIMULAÇÃO

Nesta secção é possível ao operador visualizar o formato gráfico do programa construindo no menu anterior.

Na Figura 6.11 podemos observar o layout gráfico do *Menu* Simulação.

## MENU LISTA DE PROGRAMAS

O Sistema construído têm a capacidade de armazenamento dos programas já criados anteriormente, o limite de espaço é muito elevado, permite ao operador armazenar todos os programas construídos.

Na Figura 6.12 podemos observar o layout gráfico do *Menu* Lista Programas.

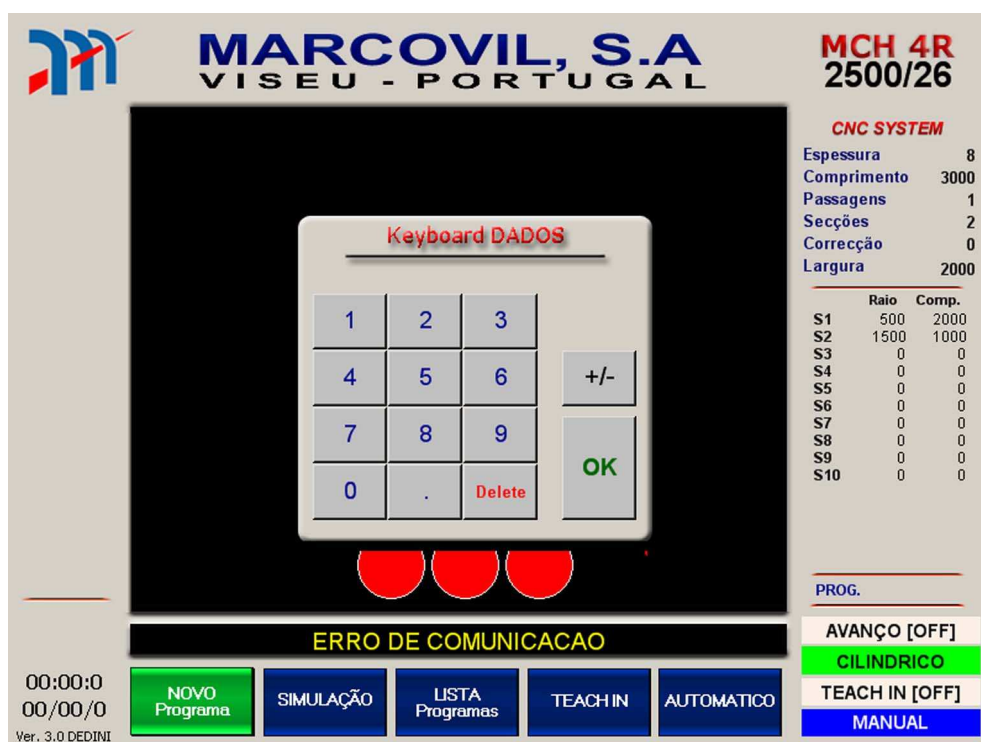


Figura 6.10: Imagem do Programa CNC MODO NOVO PROGRAMA

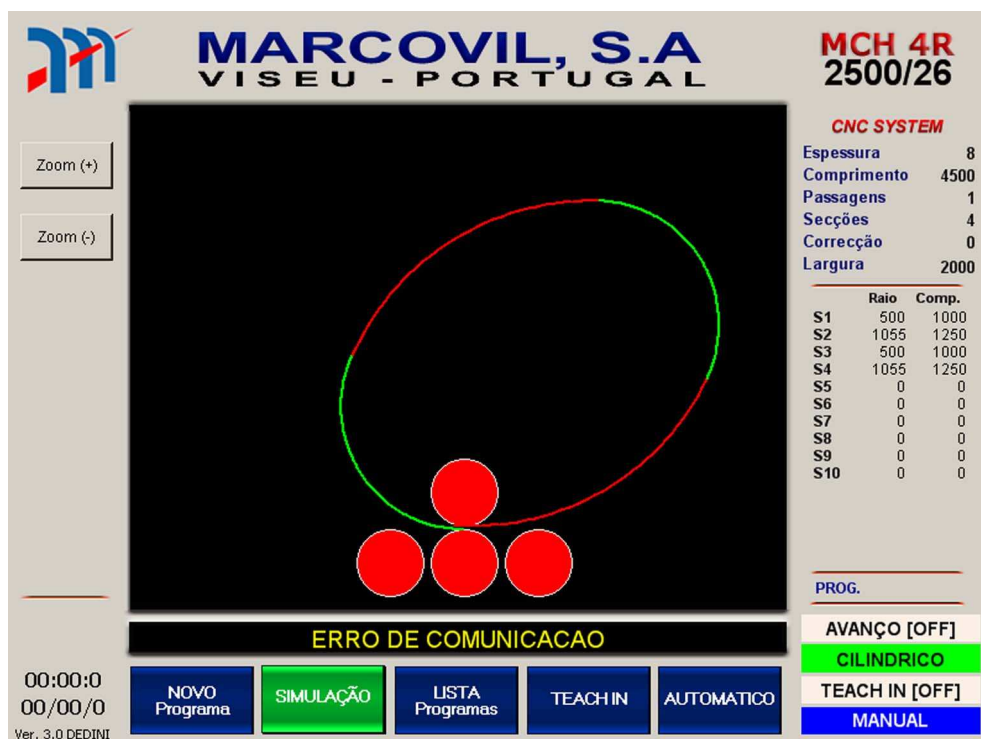


Figura 6.11: Imagem do Programa CNC MODO SIMULAÇÃO

## MENU TEACH IN

Este é uma grande utilidade existente neste comando, quando é activada esta opção o operador têm a possibilidade de construir um programa na maquina ao mesmo tempo que

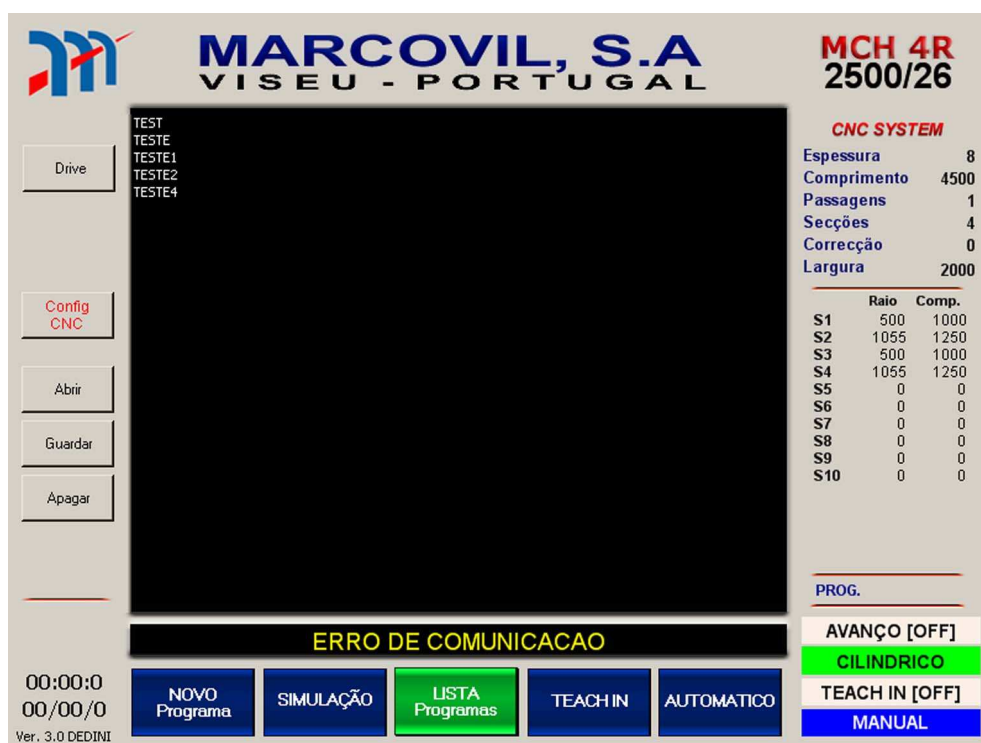


Figura 6.12: Imagem do Programa CNC MODO LISTA DE PROGRAMAS

conforma uma chapa. Este processo consiste em fazer uma memorização dos passos que o operador efectua na elaboração de uma forma cilíndrica e em seguida quando o operador o desejar a máquina efectua uma replica dos movimentos efectuados anteriormente. Esta opção de criar um programa baseado numa amostra é possível guardar no sistema de armazenamento e quando se desejar é efectuado o programa em modo automático.

Na Figura 6.13 podemos observar o layout gráfico do *Menu Automático*.

## MENU AUTOMÁTICO

Nesta secção é executado todos os programas construídos de forma directa no menu NOVO PROGRAMA ou de forma indirecta no MENU TEACH IN.

Na Figura 6.14 podemos observar o layout gráfico do *Menu Teach In*.

Em anexo irei colocar o manual de utilizador onde todo o sistema será explicado de uma maneira mas executável e onde se poderá ter a noção do tipo de equipamento que tenho relatado neste trabalho.

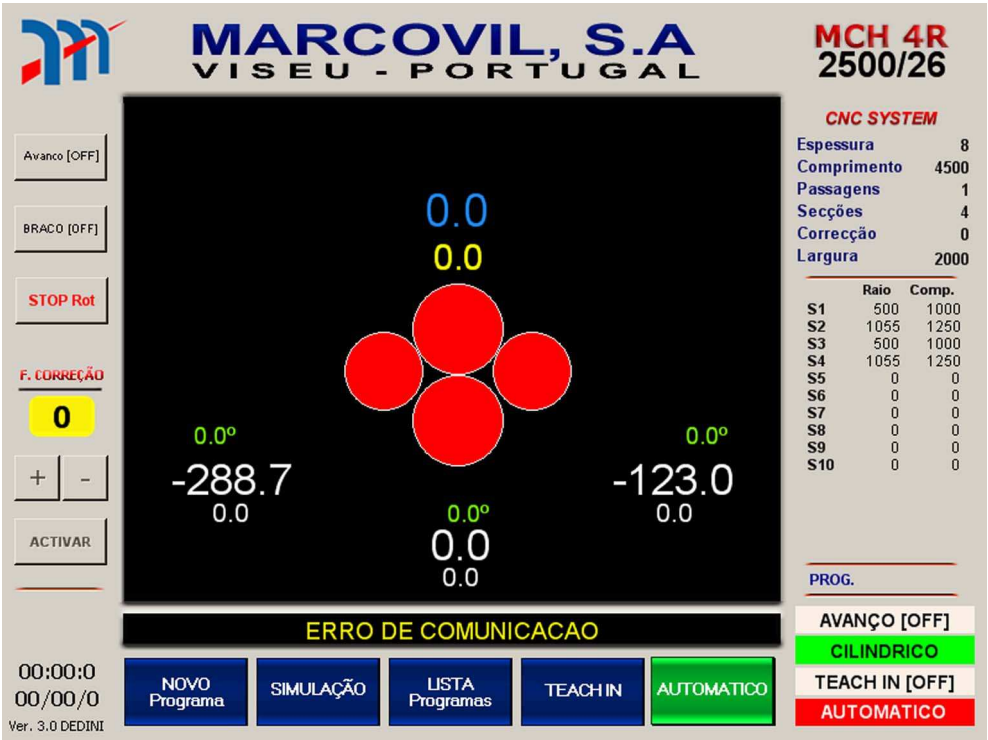


Figura 6.13: Imagem do Programa CNC MODO TEACH IN

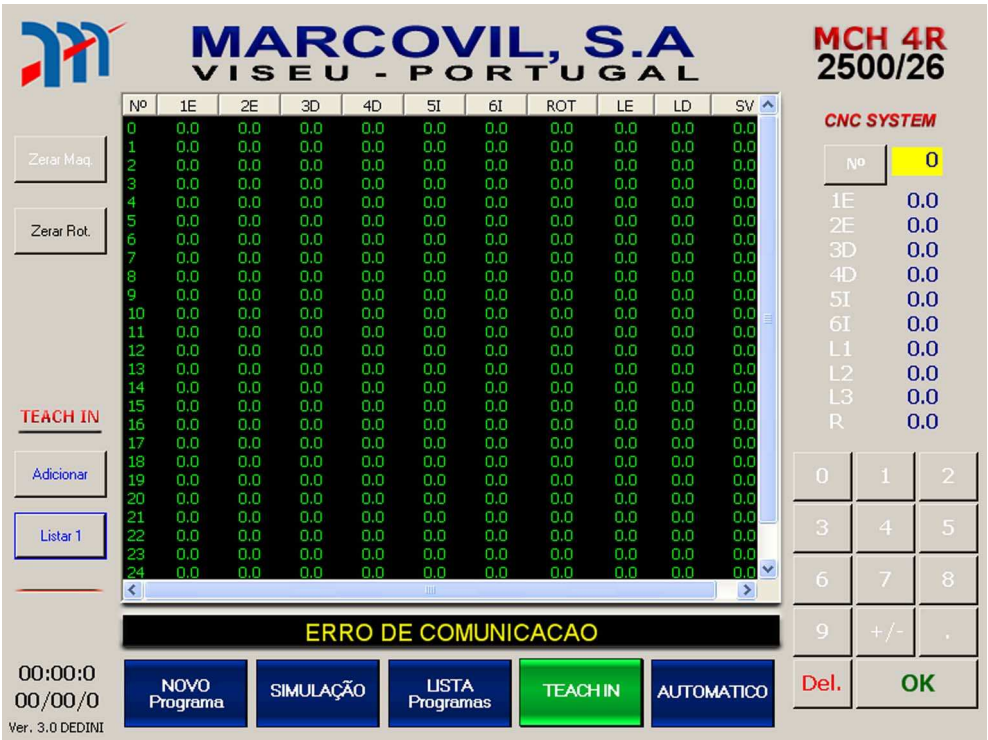


Figura 6.14: Imagem do Programa CNC MODO AUTOMÁTICO

## 6.3 Programa *PLC MASTER*

O programa de controlo do PLC Master foi desenvolvido em linguagem Ladder ou linguagem de contactos onde foi construindo o sistema de controlo manual, que acciona directamente a maquina. Foi também implementado o software de rede disponibilizado no software de programação da MELSOFT software proveniente da Mitsubishi. Os módulos de Contagem também são garantidos por funções disponibilizadas do software GX IEC Developer 7.00 onde contamos com quatro entradas rápidas de contagem por cada PLC, por este motivo implementamos dois eixos por PLC.

### 6.3.1 Diagramas de Fluxo

#### Diagrama Geral do programa PLC Master

A estrutura de software desenvolvida para o PLC Master está dividida em duas partes distintas, uma parte dedicada ao comando manual desenvolvida em linguagem de contactos, são accionados directamente pelos sinais de entrada. O outro modo é o modo de posicionamento automático, este modo está construído por funções dedicadas ao posicionamento dos vários eixos de controlo. Para uma simplicidade de trabalho e permitir um maior controlo por parte do programa principal foi criado uma estrutura em blocos de funções independentes umas das outras, assim a estrutura permite posicionar um rolo específico sem ter que obedecer a uma ordem específica.

A comunicação entre as duas redes são outro dos blocos construídos. O PLC Master detém a configuração que permite a rede entre PLCs e também faz a ponte de ligação com o programa principal. Os registos comuns mais importantes da rede de comunicação entre os PLCs são copiados para uma segunda área comum ao PLC Master e ao PC.

A rede de comunicação entre PLCs têm uma área comum onde estão 16 registos de 16 bits que servem para toda a comunicação entre PLCs, a comunicação entre o PLC Master e o programa principal trabalha com 12 registos de 16 bits e com aproximadamente 15 variáveis booleanas de um bit.

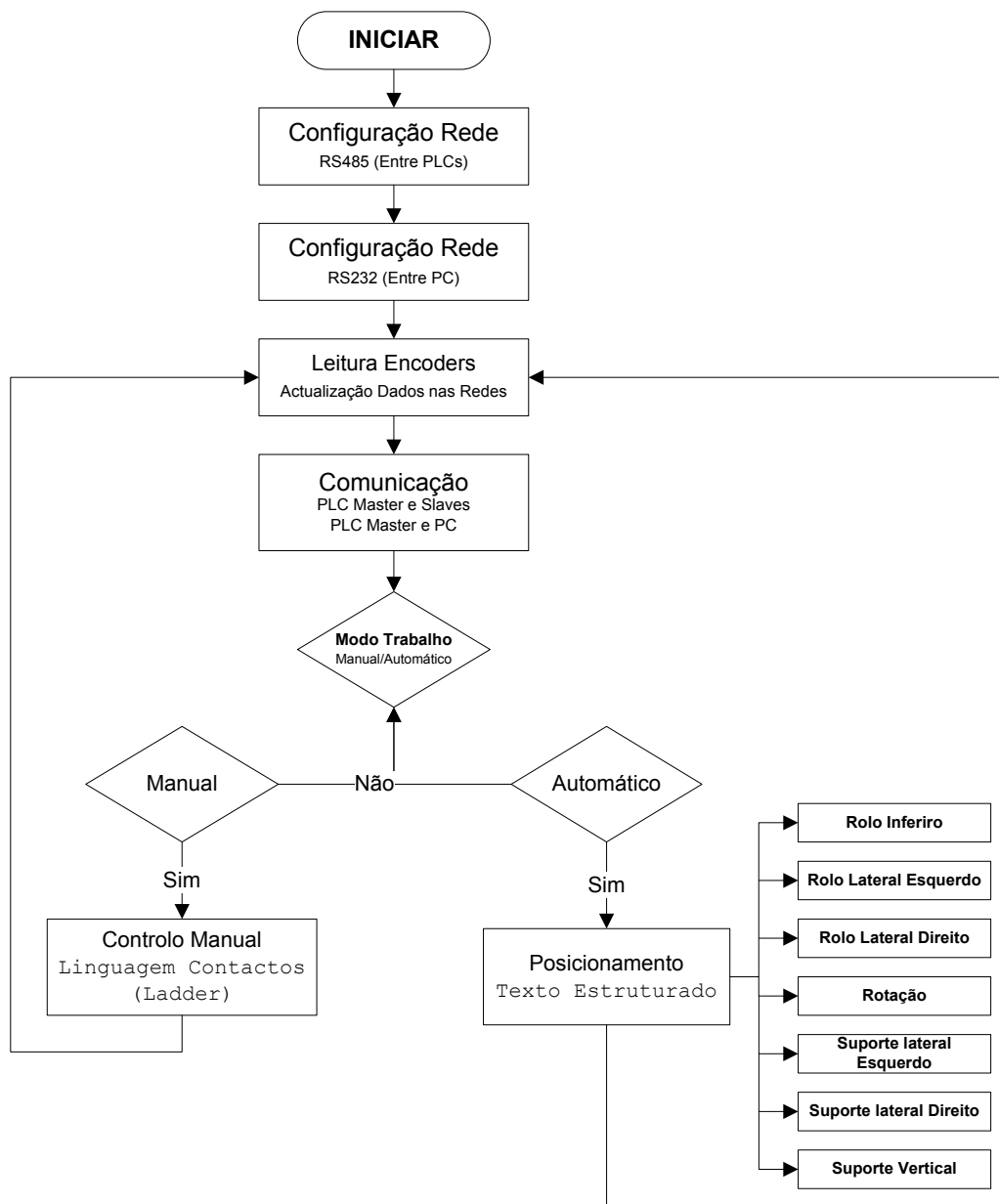


Figura 6.15: Diagrama Geral PLC Master



## Posicionamento do Rolo Inferior

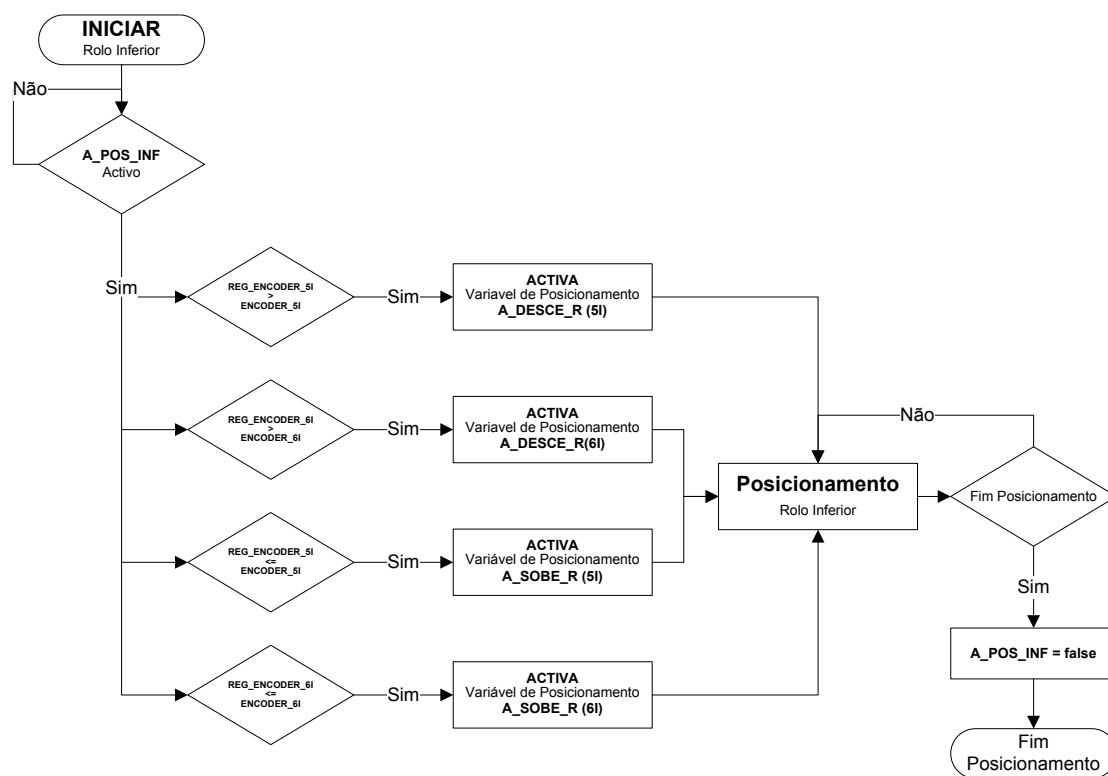


Figura 6.16: Diagrama de Posicionamento Rolo Inferior

Esta função é responsável pelo posicionamento do rolo inferior em modo automático. A execução é iniciada por uma variável de controlo A\_POS\_INF que é activada pelo programa principal *CNC\_CE.exe* alojado no PC, esta variável pertence ao programa do PLC Master. Após a activação da variável A\_POS\_INF é iniciado um ciclo de comparações entre os valores de posicionamento enviados pelo programa principal *CNC\_CE.exe* e os valores do posicionamento actual. Para realizar este processo é usado os registos de posicionamento actual que estão alojados no PLC Master, que para o rolo inferior são ENCODER\_5I para o Cilindro de eixo de controlo 5I, e ENCODER\_6I para o Cilindro do outro extremo do rolo inferior pertencente ao eixo de controlo 6I.

Os valores do posicionamento pretendido foram enviados pelo programa principal, estão armazenados em dois registos designados de REG\_ENCODER\_5I e REG\_ENCODER\_6I, quando é activado a variável de A\_POS\_INF e iniciado a comparação entre os valores dos registos e do posicionamento actual, resultante da comparação é activado variáveis (A\_DESCE\_R, A\_SOBE\_R) responsáveis por activar o movimento dos cilindros de forma a posicionarmos os cilindros em função do posicionamento pretendido.

As funções de posicionamento dos rolos laterais são executadas por funções semelhantes,



podemos observar os diagramas de fluxo na Figura 6.18 e 6.17 .

## Posicionamento do Rolo Lateral Direito

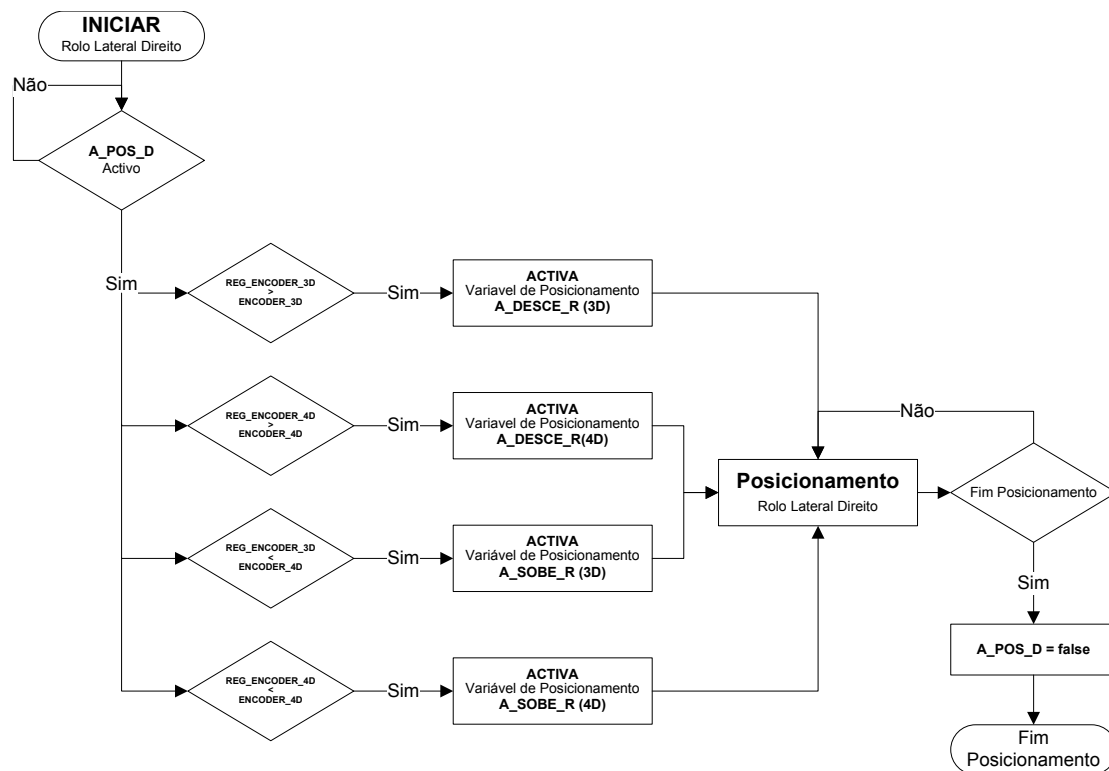


Figura 6.17: Diagrama de Posicionamento Lateral Direito

## Posicionamento do Rolo Lateral Esquerdo

### Posicionamento da Rotação

O posicionamento em modo automático do eixo de controlo da rotação é semelhante aos outros eixos de controlo. É feita a activação da função de posicionamento pelo programa principal, após o início da função será comparado o valor actual do eixo de controlo da rotação com o valor armazenado no registo definido para guardar o valor enviado pelo programa principal, após a comparação será accionado a rotação no sentido directo ou inverso dependendo do resultado da comparação.

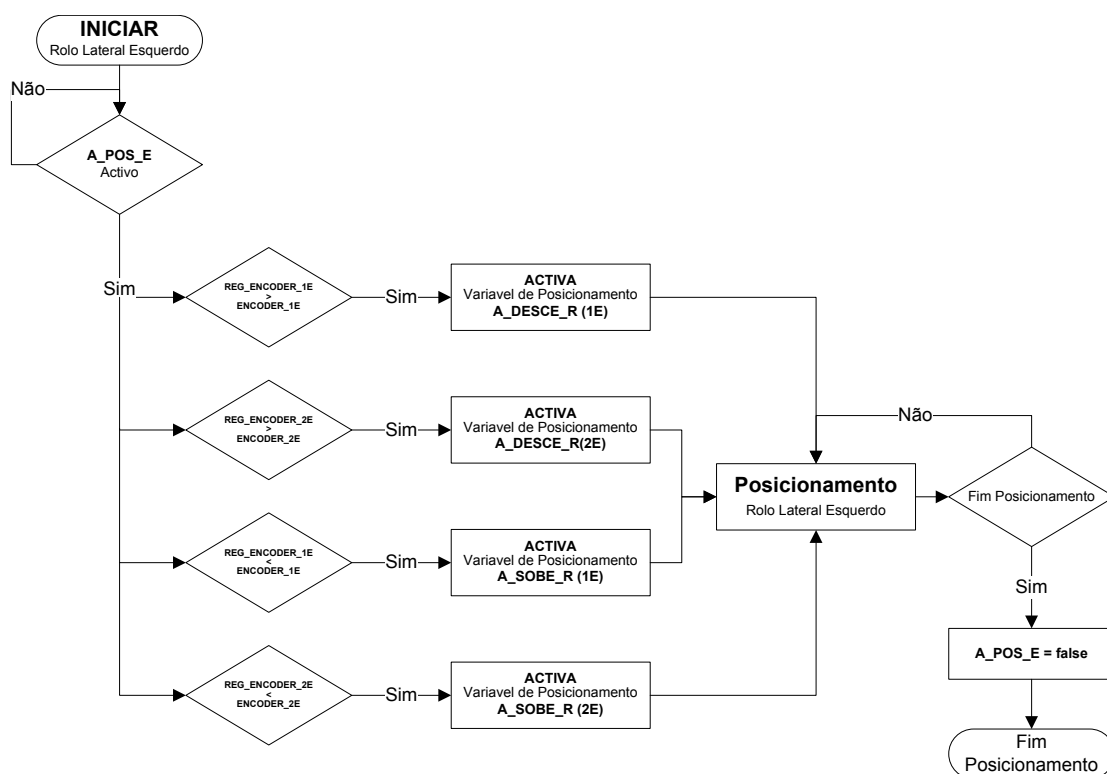


Figura 6.18: Diagrama de Posicionamento Lateral Esquerdo

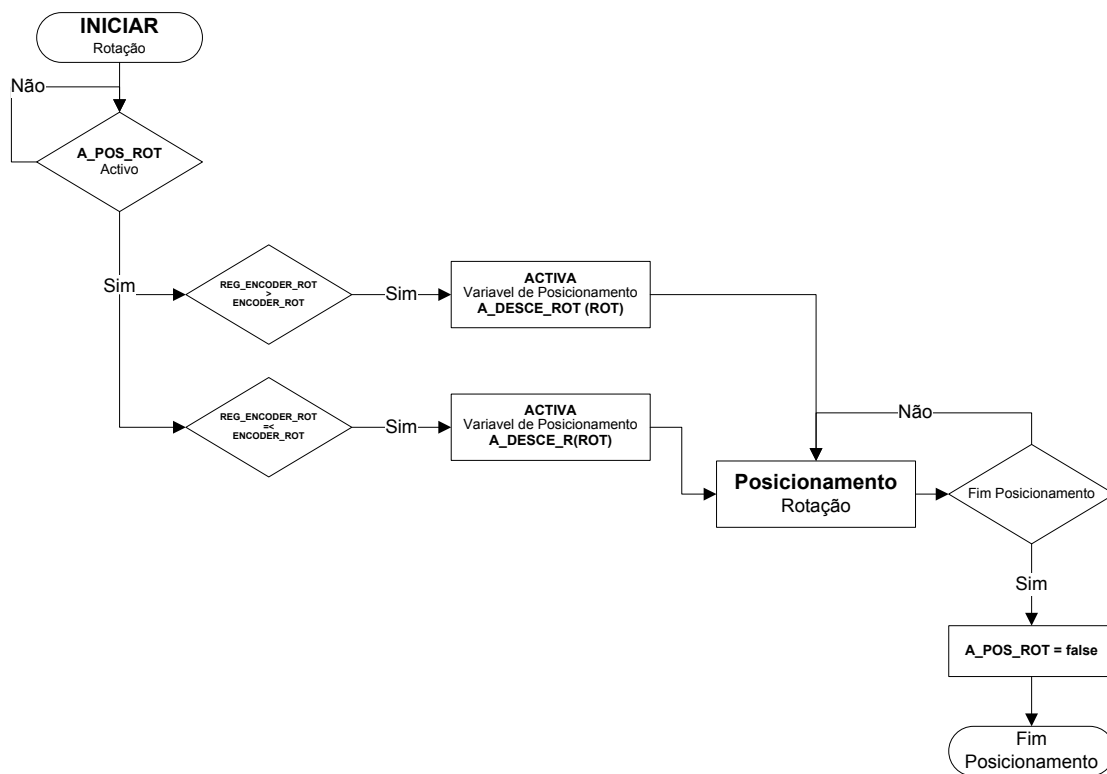


Figura 6.19: Diagrama de Posicionamento Rotação

## 6.4 Programa *PLC SLAVE*

O Software dos PLCs SLAVE foi desenvolvido também no GX IEC Developer 7.00 onde foi implementado o software de ligação á rede de comunicação, e com base das quatro entradas rápidas foi implementado o sistema de leitura dos eixos. o controlo de posição é feito com o uso de dois registos de posição, um com a posição actual do eixo e outro com a posição pretendida, neste ultimo registo é colocado o valor pelo PC Industrial este registo é um registo de rede, assim quando se quer posicionar o eixo na posição que se encontra no seu registo de posição activamos um bit para iniciar o posicionamento, quando o posicionamento é atingido é activado outro bit de sinalização.



## 6.5 Trabalho Futuro

Este trabalho de desenvolvimento não se encerra aqui, existem neste momento o objectivo de continuar a adicionar módulos de forma a tornar o sistema de controlo cada dia mais perfeito. Um dos objectivos para os próximos doze meses será introduzir uma malha de feedback que nos possa auxiliar no registo do desvio que está a ser provocado em relação ao objectivo programado, para que se possa introduzir um sistema de correcção da conformação da chapa. Esta malha já existe para o posicionamento dos eixos mas no comum das maquinas é suficiente, neste tipo de maquina não é suficiente pelo facto da natureza da chapa interferir no processo de conformação. Outro objectivo é implementar no sistema de controlo a execução de cones em modo automático, situação que não ocorre devido ao escorregamento da chapa causado pelo menor aperto exigido.



# Conclusões

No decorrer deste trabalho foi verificado que a aposta numa topologia de sistema de controlo distribuído é vantajosa para um sistema de comando que está em plena evolução. Foi definido como sistema operativo base o Windows CE 5.0 que verificou na pratica uma melhoria no rendimento do PC Industrial. A deslocalização das tarefas de controlo do PC Industrial para os PLCs com a consequência directa da aplicação do sistema distribuído resulta na diminuição do peso PC Industrial para o funcionamento da calandra e um aumento de potencialidade na execução de multitarefas de posicionamento, em caso de falha da aplicação CNC\_CE.exe a maquina continua funcional, claro sem a potencialidade do sistema CNC. Quanto á fiabilidade do sistema, não foram detectadas anomalias dignas de registo. A aceitação deste novo comando na mercado mundial têm sido muito boa, as presenças nas feiras internacionais, estou a referir a maior feira da especialidade na Europa (EUROBLECH) e a maior também da América Latina (CORTE E CONFORMAÇÃO), tem tido resultados em vendas muitos bons. A adaptação dos algoritmos de posicionamento e a adaptação do processo de calandragem deram um impulso muito grande nos resultados de conformação em automático. Quando iniciei ha três anos este projecto de desenvolvimento, em percentagem a Marcovil vendia por ano 10% das maquinas equipadas com sistema CNC, nos dias de hoje a Marcovil,S.A exporta 10% das Maquinas sem sistema CNC.

A comparação deste comando com outros comandos de marcas concorrentes no mercado mundial das calandras, é muito satisfatória para o nosso lado, somos reconhecidos pela concorrência e observados em cada feira onde nos apresentamos, sinal que este trabalho têm sido reconhecido.





# Bibliografia

- [1] Eng. Arivelto Bustamante Fialho, *Automação Hidráulica*, 3ª Edição, Érica, 2006.
- [2] António M. S. Francisco, *Autómatos Programáveis*, Lidel, 2007.
- [3] João R. Caldas Pinto, *Técnicas de Automação*, Edições Técnicas e Profissionais, 2007.
- [4] Carlos Relvas, *Controlo Numérico Computorizado*, Publindústria, 2002.
- [5] J. Norberto Pires, *Automação Industrial*, Edições Técnicas e Profissionais.
- [6] Simon Robinson, Christian Nagel, Jay Glynn, Morgan Skinner, Karli Watson, Bill Evjen, *Professional C#, 3ª Edição*, Wiley Publishing Inc.
- [7] Rockford Lhotka, *Expert C# 2005 Business Objects*, 2ª Edição, Apress.
- [8] Tom Archer, *Inside C#*, McGraw Hill, 2002.
- [9] Vítor Pereira, *O Guia Prático do Visual C# 2005 Express*, Centro Atlantico, 2006.
- [10] *User's Manual Fx Communication (RS232C, RS485, RS422)*, Versão E, Manual Mitsubishi, 2003.
- [11] *Programming Manual FX0S/FX0N/FX2N series*, Manual Mitsubishi, 2003.
- [12] *Communications Manual FX1S/FX1N/FX2N(C)/FX3U*, Manual Mitsubishi, 2004.
- [13] *Programming Manual Basic & Applied FX3U (FX3UC)*, Manual Mitsubishi, 2007.
- [14] *Hardware Manual FX3U (FX3UC)*, Manual Mitsubishi, 2007.
- [15] *Operating Manual GT Designer2*, Manual Mitsubishi, 2004.
- [16] *Reference Manual GX IEC Developer 7.01*, Manual Mitsubishi, 2006.



## **MANUAL DE UTILIZAÇÃO DO COMANDO**



# **CNC SYSTEM MARCOVIL**



## ÍNDICE

I.	ÍNDICE .....	1
II.	CNC MARCOVIL .....	4
III.	CONSOLA MARCOVIL.....	5
IV.	INSTRUÇÕES DE UTILIZAÇÃO.....	6
▪	INICIAR O CNC MARCOVIL .....	6
V.	PROGRAMAÇÃO.....	7
▪	INTRODUZIR DADOS PARA UMA SECÇÃO E FECHAR O RAIOS.....	8
▪	INTRODUZIR DADOS PARA UMA SECÇÃO E UM RAIOS .....	9
▪	INTRODUZIR DADOS PARA VARIAS SECÇÕES E RAIOS .....	9
▪	MENSAGENS DE ERROS NA PROGRAMAÇÃO .....	10
VI.	PROGRAMAS GUARDAR APAGAR CARREGAR.....	11
▪	GRAVAR PROGRAMAS .....	11
▪	CARREGAR PROGRAMAS .....	12
▪	APAGAR PROGRAMAS .....	12
VII.	EXECUÇÃO DE PROGRAMAS.....	13
▪	AVANÇO [OFF].....	14
▪	AVANÇO [ON] .....	14
▪	BRAÇO AUTOMATICO [OFF] .....	15
▪	BRAÇO AUTOMATICO [ON].....	15
VIII.	TEACH IN .....	15
▪	MODO DE ACTIVAÇÃO “TEACH IN” .....	15
▪	MODO DE TRABALHO TEACH IN.....	16
▪	ALTERAÇÃO DE UM VALOR REGISTADO .....	19
▪	GUARDAR PROGRAMA TEACH IN.....	22
▪	ABRIR PROGRAMA TEACH IN .....	23
IX.	CALANDRAGEM DE PEÇAS CÓNICAS .....	26
X.	PROGRAMAÇÃO DE VIROLAS CILÍNDRICAS.....	27

## **II. CNC Marcovil**

Sendo a **Marcovil**, S.A. uma empresa de objectivos sempre arrojados, com uma filosofia de apresentar os seus produtos garantidamente fiáveis, eficazes e de engenharia surpreendentemente simples, não seria de modo diferente o desenvolvimento e engenho de um sistema de controlo numérico computadorizado. Assim o **CNC Marcovil**. é também um produto de grande qualidade, extrema facilidade de manuseamento, indo ao encontro de uma interligação homem-máquina como de homem-homem se tratasse, tendo por base um sistema de arquitectura PC instalado em moldura industrial optimizado para operar em ambientes de extrema poluição atmosférica e extremamente tolerante a pavimento irregulares, choques violentos e quedas accidentais.

O sistema de controlo numérico computadorizado **Marcovil**. Tem como base um computador PC que comanda um arrojado e inovador conjunto electrónico de interligação com os meios mecânicos convencionais nomeadamente cilindros hidráulicos.

De ampla visão e fácil acesso, encontra-se instalado na face vertical da consola móvel de comandos da máquina.



### III. CONSOLA MARCOVIL



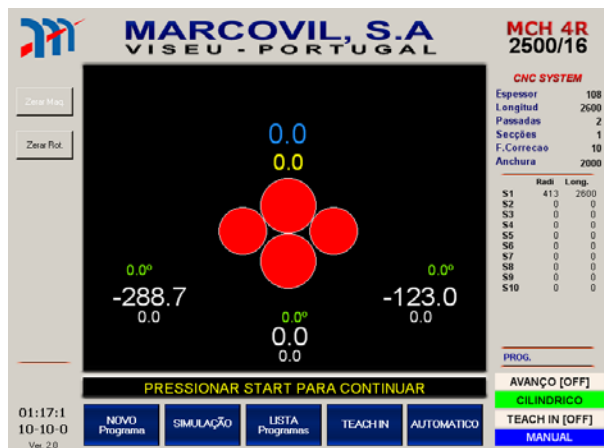
- 1 - Interruptor - **EMERGÊNCIA**
- 2 - Botão ligar CNC - Sistema Hidráulico
- 3 - Botão desligar CNC - Sistema Hidráulico
- 4 - Botão INICIO de Programa
- 5 - Botão pressão - Rotação dos rolos à esquerda
- 6 - Botão STOP de Programa
- 7 - Interruptor Rotação Esquerda/Direita
- 8 - Botão pressão - Rotação dos rolos à direita
- 9 - Accionador Sobe/Desce - Suporte Lateral 2
- 10 - Accionador Sobe/Desce - Rolo Lateral 1
- 11 - Accionador Sobe/Desce - Rolo Inferior
- 12 - Accionador Sobe/Desce - Rolo Lateral 2
- 13 - Interruptor ligar CNC - Consola de Comandos
- 14 - Interruptor de Modo Cones/Cilíndrico
- 15 - Interruptor de Rotação Lenta e Normal
- 16 - Interruptor ABRE/FECHA - Braço do rolo superior
- 17 - Interruptor Sobe/Desce Suporte Vertical  
,Sobe/Desce Suporte Lateral 1

A Consola de Comandos, no modo **Manual** controla todas as funções dos botões e accionadores como se não tivesse sistema **CNC**.

## IV. INSTRUÇÕES de UTILIZAÇÃO

### ▪ INICIAR O CNC MARCOVIL

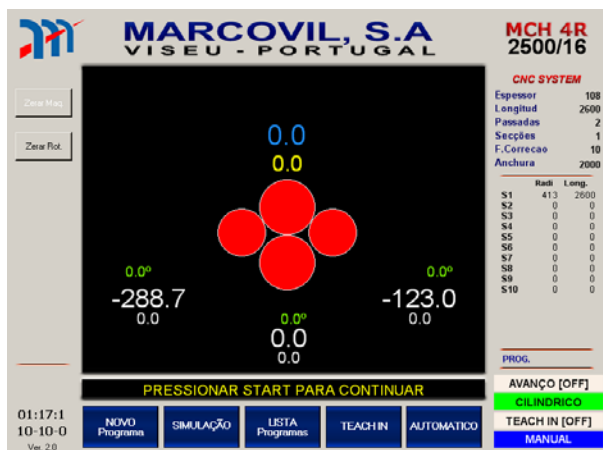
O *CNC Marcovil* inicia-se automaticamente sempre que se acciona o **Botão 13**, demorará algum tempo ate estar completamente em forma de trabalho. Quando o sistema estiver pronto para trabalhar mostrará a seguinte figura no ecrã.



Após o início do *CNC Marcovil* o sistema esta no modo de trabalho **Manual**, accionando os botões de pressão e interruptores existe a resposta hidráulica por parte da maquina tendo assim todo o controlo **Manual**.

Depois de accionar o sistema hidráulico (pressionando o **Botão 2**, da consola de comandos), o *CNC Marcovil* completará o arranque do computador e ficará em modo **Manual**. Em modo **Manual**, accionar primeiro o rolo inferior, através do **Accionador 11** da consola, na posição de subida até à sua posição encostado ao rolo superior. Accionar de seguida o rolo lateral 1, através do **Accionador 10** da consola, na posição de subida até à posição de encosto ao rolo superior.

Repetir a operação com o rolo lateral 2 (direita). Com o rolo inferior e os rolos laterais encostados ao rolo superior, pressionar no ecrã a **Botão ZERAR**. Os contadores do écran central ficam a Zero. Antes de iniciar qualquer programa esta condição tem que ser observada.

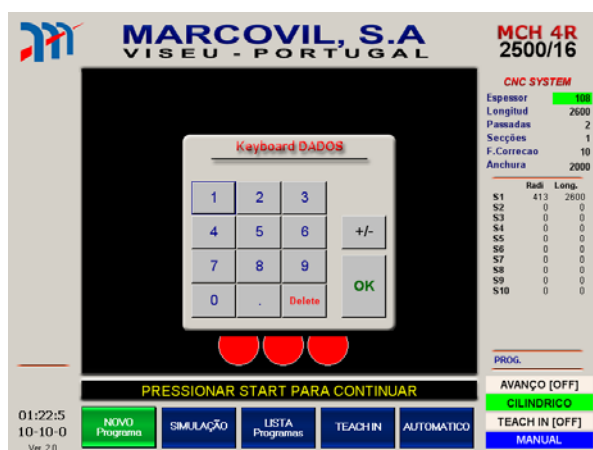




## V. PROGRAMAÇÃO

Pressionar o **Botão DADOS**: entra-se no menu de **DADOS** mostrará uma tabela com os campos **ESPESSURA, COMPRIMENTO, PASSAGENS, SECÇÕES, FACTOR ROLO OPOSTO, LARGURA**. Ao introduzir valores no teclado inicia a colocação do valor do campo **ESPESSURA**, sempre que pressionar a tecla ENTER irá passar para um novo campo assim sucessivamente até ao campo **LARGURA**. Se pretender voltar a traz para alterar um valor basta pressionar no botão correspondente ao campo a alterar, depois é só introduzir o valor pelo teclado e pressionar a tecla **ENTER** para transferir o valor do teclado para o campo a alterar.

No final de introduzir o valor do campo largura será feito o validar dos dados principais, se ocorrer um erro irá surgir uma informação de erro na caixa de texto da direita e assinaladas a vermelho os campos com erros. Para corrigir os erros basta seleccionar os campos com erro pressionando o botão correspondente e introduzir novos valores e voltar a correr os campos restantes até á **LARGURA** e validar de novo os dados principais.



Assim que se valide os dados principais, surgirá os dados secundários (dados do comprimento e raio de cada secção) introduzindo da mesma forma o **COMPRIMENTO** e **RAIO** de cada Secção sendo a soma dos Comprimentos de cada secção igual ao Comprimento total.

No fim de introduzir o valor do raio da última secção será validado todos os valores, se todos os valores estiverem dentro dos limites será feita a validação final e surgirá a activo o **BOTÃO OK** (Este botão está sempre desactivo só e activo quando se introduz o valor do raio da ultima secção e feito a validação se o resultado da validação for positiva irá activar o **BOTÃO OK** passando a ser visível (*localiza-se atrás do teclado como também o botão EXIT*) este botão e o que possibilita a saída do menu **DADOS** com os novos dados programados, se desejarmos sair e manter os dados anteriores basta pressionar o **Novo Programa**.

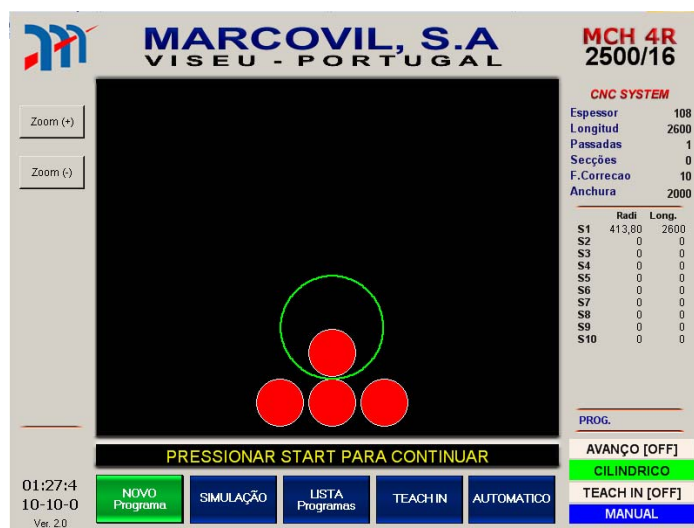
## ▪ INTRODUIZIR DADOS PARA UMA SECÇÃO E FECHAR O RAIO



Os dados desta forma com o campo **SECÇÕES = 0**, significa que teremos um raio fechado. Os valores do RAIO e visível no final do OK do teclado e mostra na parte direita o raio calculado para fechar o comprimento de chapa metálica inserida na **COMPRIMENTO** de forma a termos um cilindro.

Os dados programados ficam visíveis na tabela da na coluna da direita que tem o nome de **DADOS**, os valores dos raios e das secções estão abaixo do nome **DADOS OF SECÇÕES**.

Para observar a geometria dos dados introduzidos, depois de termos saído do menu **DADOS**, pressionamos o **BOTÃO SIMULAR** e podemos observar a geometria dos dados programados.



Dentro do menu **SIMULAR** podemos observar a forma dos dados programados, podemos fazer zoom pressionado o **BOTÃO -** e **+**, para sair do menu SIMULAR pressiona-se o **BOTÃO SAIR**.

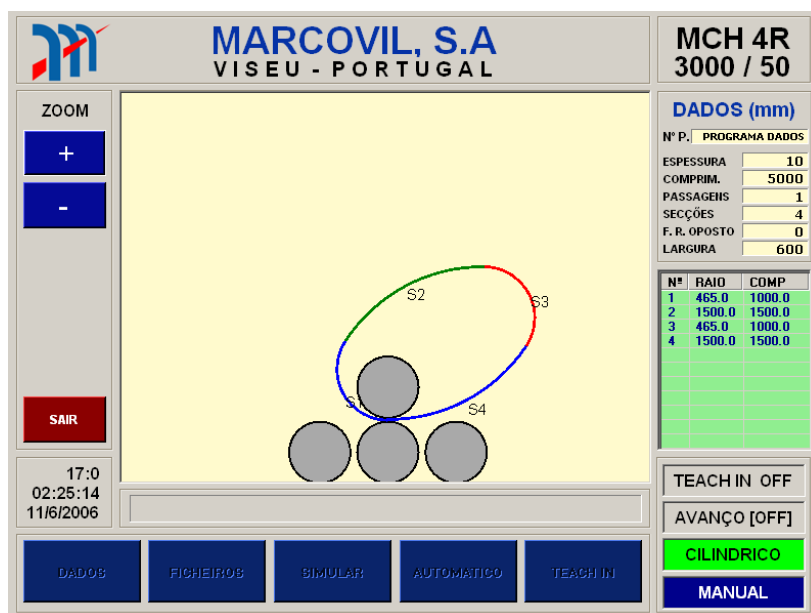
### ▪ INTRODUIZIR DADOS PARA UMA SECÇÃO E UM RAIOS

Para termos uma secção com um raio predefinido, introduzimos na menu **DADOS** na caixa **SECÇÕES** o valor de 1, depois de introduzir os outros dados, temos do lado direito duas caixas uma de comprimento que corresponde a **COMPRIMENTO** e outra onde colocamos o valor do **RAIO** pretendido, o **COMPRIMENTO** tem de ser igual ao comprimento introduzido anterior, no final pressionamos o **BOTÃO de OK** do teclado, depois pressionar **BOTÃO OK** para sair do menu **DADOS**.

### ▪ INTRODUIZIR DADOS PARA VARIAS SECÇÕES E RAIOS

Para termos varias secção com vários raios predefinidos, introduzimos no menu **DADOS** na caixa **SECCOES** o valor do número de secções, depois de introduzir os outros dados irá ser feita a validação dos valores principais e depois será necessário introduzir os valores do **RAIO** e do **COMPRIMENTO** de cada secção e no final pressionar o **BOTÃO OK** para sair do menu **DADOS**.

O valor de zero (0) no **RAIO** de uma secção corresponde a um raio nulo ou seja a uma superfície plana.



## ▪ MENSAGENS DE ERROS NA PROGRAMAÇÃO

Quando da programação de um programa existem um número de erros que podem ocorrer.

**ERRO DE ESPESSURA** – O valor programado têm que estar abaixo da relação para a espessura máxima.

$$\text{Ex. } \frac{Largura}{LarguraN} \times Espessura \leq EspessuraN$$

**ERRO PASSAGENS** – O Valor deve estar entre 1 e 7

**ERRO SECÇÕES** – O Valor entre 0 e 10;

**ERRO F. Rolo Oposto** – Valor entre -40 a 40;

**ERRO LARGURA** – Valor inferior a largura nominal;

**ERRO DE COMPRIMENTO TOTAL** – A soma dos Comprimentos da Secções têm de ser igual ao Comprimento

**ERRO DE RAO DA SECÇÃO** – Raio têm de ser maior que 1.15 do Raio do Rolo Superior.

## VI. PROGRAMAS GUARDAR APAGAR CARREGAR

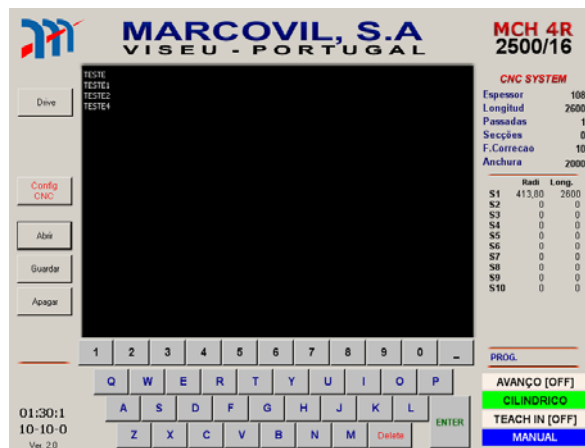
### ▪ GRAVAR PROGRAMAS

Depois de introduzir os dados como descrito, pressionando o **BOTÃO FICHEIRO**, o ecrã mostra o menu **FICHEIRO** onde se podem ver os **Programas existentes**.



Para guardar os valores introduzidos no menu **DADOS** num programa, pressione o **BOTÃO GRAVAR**, surge um teclado para introduzir o nome do ficheiro, pressionado as teclas damos o nome ao programa no final pressionamos o **BOTÃO OK** do **TECLADO** e surge no campo central o nome do programa armazenado.

Para guardar os valores no disco DISK FLASH temos que antes de pressionar no **BOTÃO DISK FLASH** e depois no **BOTÃO GRAVAR**.



## ▪ CARREGAR PROGRAMAS

Para carregar programas guardados basta pressionar o **BOTÃO OPEN** de seguida surge um menu com um teclado onde introduzimos o nome que pretendemos abrir e depois pressionamos o **BOTÃO OK** do teclado e o programa é carregado. Para ver os dados do programa pressionamos o **BOTÃO SAIR FICHEIRO** e pressionamos no **BOTÃO DADOS** podemos ver os dados do ficheiro, se pressionamos o **BOTÃO SUMULAR** vemos a forma geométrica dos dados.



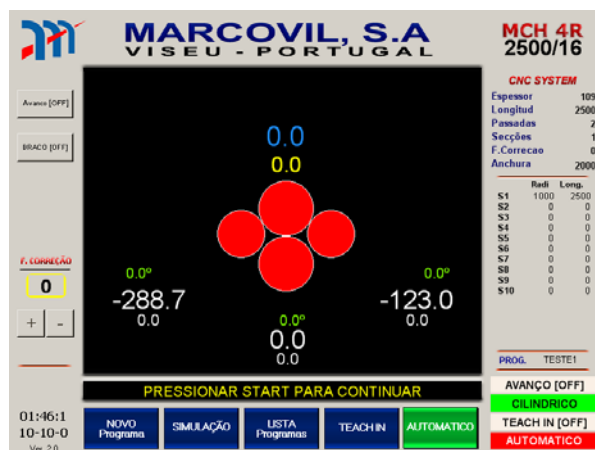
## ▪ APAGAR PROGRAMAS

Para apagar um programa guardado basta dentro do menu FICHEIRO pressionar o **BOTÃO DELETE**, surge a mensagem NOME DO FICHEIRO, seleccionado a caixa de texto que surge de frente da mensagem NOME DO FICHEIRO, surge o teclado do lado direito, marcado o nome do programa para

apagar e depois pressionar o BOTÃO OK do teclado, surge a mensagem apagado o ficheiro com sucesso e desaparece da listagem de programas.

## VII. EXECUÇÃO DE PROGRAMAS

Pressionando o **BOTÃO AUTOMÁTICO**, o modo de operação passa a **Automático**. Pressionando agora o **Botão 4-Início** da consola móvel, o **CNC Marcovil** executará o programa introduzido ou seleccionado.



Depois de pressionar o **Botão 4 INÍCIO**, da consola móvel, o **CNC** acciona o braço de suporte do rolo superior na posição de fechado, posiciona de imediato o rolo inferior de acordo com a espessura da chapa programada, seguindo-se o posicionamento do rolo lateral 1 (lado esquerdo) e o posicionamento do rolo lateral 2 (lado Direito).

O **CNC** pára nesta altura e uma mensagem surge no fundo do ecrã: **INTRODUZA CHAPA**; introduzir a chapa que se pretende calandrar, para que a mesma fique encostada ao rolo lateral 2.

É importante que a chapa fique encostada em toda a largura, pois isso é determinante para uma calandragem perfeita. Depois de introduzida a chapa pressione o **Botão 4 INÍCIO**, da consola de comandos.

O **CNC** posiciona o rolo inferior de forma a aprisionar a chapa contra o rolo superior e seguidamente rola os rolos à esquerda, posicionando a chapa para a primeira passagem de calandragem.

No fundo do ecrã surge uma mensagem: **PRESSIONE START PARA CONTINUAR**.

Ao pressionar o **Botão 4 INÍCIO**, da consola de comandos, o **CNC** reinicia o programa de calandragem, baixando ou subindo os rolos, fazendo rolar a chapa com paragens e subidas de rolos, de acordo com o sistema, até finalizar a primeira passagem, ao fim da qual a mensagem **PRESSIONE START PARA CONTINUAR**, surge no fundo do ecrã.

Pressionando **Botão 4 INÍCIO**, sempre que a máquina finalize uma operação de calandragem, chegará ao fim do programa.

O **CNC** automaticamente controla a descida do rolo inferior, rolos laterais e abertura do braço de suporte do rolo superior, depois de se pressionar no botão nº4 **INÍCIO**, sempre que a mensagem **PRESSIONE START PARA CONTINUAR**, aparecer no fundo do ecrã. No final da calandragem os rolos laterais e inferior baixam e o braço de suporte abre.

Para iniciar novamente o programa em automático basta pressionar o botão nº4 **INÍCIO**.

#### ▪ **AVANÇO [OFF]**

Neste modo de operação o **CNC** executará o programa de forma faseada, parando cada vez que uma passagem esteja realizada. Pressionado o **Botão 4-Início** na consola móvel, reinicia-se o programa.

Para activar o modo **AVANÇO [ON]** e o **CNC** tenha um processo sem paragem basta pressionar o **BOTÃO AVANÇO**.

#### ▪ **AVANÇO [ON]**

Pressione o **BOTÃO AVANÇO** no menu **AUTOMATICO**; se o **CNC** estiver no modo **AVANÇO [ON]** a mensagem mudará para **Avanço [ON]**. Isto significa que o programa será executado pelo **CNC** sem interrupção só pára quando chegar ao final do programa ou se for activado de novo o **BOTÃO AVANÇO** passando o modo para **AVANÇO [OFF]**.

As pausas feitas no modo **AVANÇO [OFF]** são úteis para posicionar os suportes laterais e vertical de auxílio á chapa.

Outras mensagens surgirão no ecrã de acordo com as operações em curso:

**ROTAÇÃO ROLOS DIREITA** – Rolos com movimento giratório à direita

**ROTAÇÃO ROLOS ESQUERDA** – Rolos com movimento giratório à esquerda

**SUBIDA ROLO INFERIOR** – Rolo inferior sobe

**DESCIDA ROLO INFERIOR** – Rolo inferior desce



**SUBIDA ROLO LATERAL 1** – Rolo lateral 1 sobe  
**DESCIDA ROLO LATERAL 1** – Rolo lateral 1 desce  
**SUBIDA ROLO LATERAL 2** – Rolo lateral 2 sobe  
**DESCIDA ROLO LATERAL 2** – Rolo lateral 2 desce.

- **BRAÇO AUTOMATICO [OFF]**

Esta função desactivada ou seja o BOTÃO ter a mensagem Braço Automático [OFF] implica que no modo de funcionamento automático do CNC na posição final não será aberto o suporte do rolo superior para retirar a chapa.

- **BRAÇO AUTOMATICO [ON]**

Com o modo Braço Automático [ON] o ultimo posicionamento do processo automático do CNC será baixar todos os rolos principais e abrir o suporte do rolo superior para retirar o trabalho calandrado.

## VIII. *TEACH IN*

O Modo **TEACH IN** é uma forma prática de o operador construir os seus próprios programas baseando-se no processo manual.

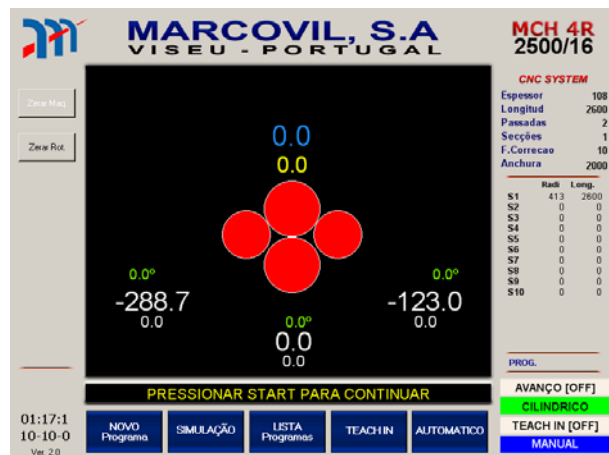
Trabalhando em **MODO MANUAL** e ao iniciar a execução de um trabalho terá uma oportunidade de construir um programa que mais tarde executado em **MODO AUTOMATICO** terá o mesmo resultado do resultado da execução em **MODO MANUAL**, podendo ser guardado e executado milhares de vezes.

Este modo baseia-se no registo de todas as posições dos rolos principais e secundários e também da rotação. Este processo é construído fase a fase ou seja de passagem em passagem, sendo em cada passagem feito o registo de cada posição dos rolos principais e secundários numa tabela, que no final terá o registo de todas as posições ocorridas na execução no trabalho em **MODO MANUAL**.

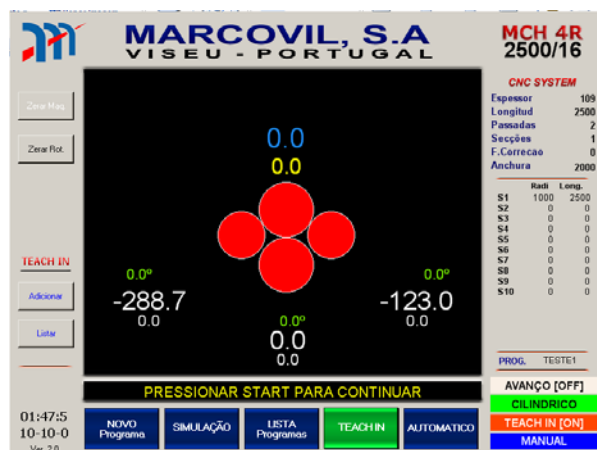
No fim da execução do processo em **MODO MANUAL** existe a possibilidade de execução directa em **MODO AUTOMATICO** do programa construído, ou também a possibilidade de alterar os valores capturados, e futuramente guardar o programa construído dando-lhe um nome para que possa ser usado mais tarde.

- **MODO DE ACTIVAÇÃO “TEACH IN”**

Em primeiro lugar vamos trabalhar em **MODO MANUAL**



No **MODO MANUAL** pressionamos o **BOTÃO TEACH IN**, activando o **MODO TEACH IN**, de seguida surgirá a mensagem **TEACH IN [ON]** no canto inferior direito e também irá surgir dois novos **BOTÕES** no menu lateral esquerdo, **BOTÃO GUARDAR POSIÇÃO <TEACH IN>** e **EDITAR <TEACH IN>**.



#### ▪ MODO DE TRABALHO TEACH IN

Ao ter o **MODO TEACH IN** activo podemos iniciar o processo de fabrico e a mesmo tempo construir um programa baseado no processo manual.

Iniciamos o processo de calandragem mas em cada passagem vamos pressionar no **BOTÃO GUARDAR POSIÇÃO <TEACH IN>** para que seja possível registar a posição de todos os rolos, para que seja possível mais tarde posicionar a fase actual.

Uma passagem entende-se por um máximo de um posicionamento diferente por rolos seguindo a ordem apresentada no próximo parágrafo.

Um exemplo:

- 1º- Posiciona Rolo Esquerdo
- 2º- Posiciona Rolo Inferior
- 3º- Posiciona Rolo Direito
- 4º- Posiciona Suportes Laterais
- 5º- Posiciona Suportes Verticais
- 6º- Posiciona Rotação

No fim desta sequência tenho que pressionar o **BOTAO GUARDAR POSIÇÃO <TEACH IN>** para que seja possível registar a posição de todos os rolos.

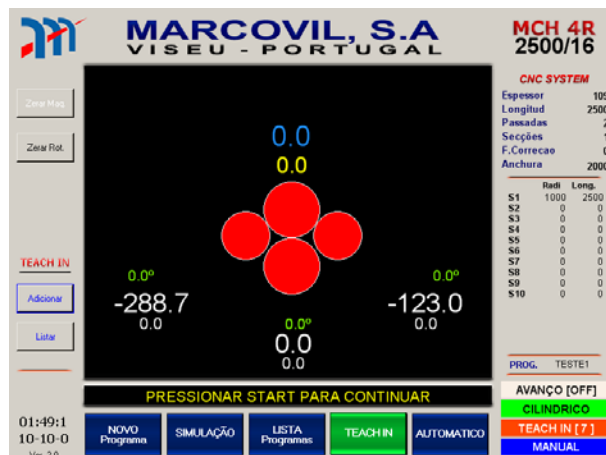
A ordem de posicionamento tem que ser respeitada, não é necessário posicionar todos os rolos, basta posicionar os que estão a interferir no processo de calandragem mas respeitando sempre a ordem assim que atingimos o **POSICIONAMENTO DE GRAU MAIS ELEVADO DO PROCESSO** temos que pressionar o **BOTAO GUARDAR POSIÇÃO <TEACH IN>** para que seja possível registar a posição de todos os rolos.

Existem um **MAXIMO de 100 PASSAGENS** (Numero muito grande para um processo de fabrico)

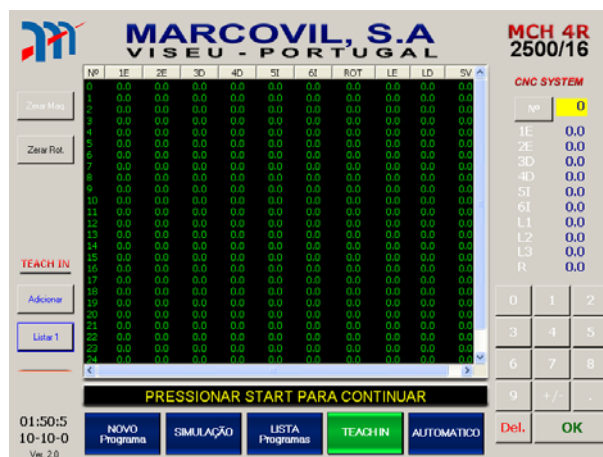
De um modo simples o posicionamento em **MOD0 AUTOMATICO** com **MOD0 TEACH IN [ON]** dos rolos terá a seguinte ordem:

- 1º- Posiciona Rolo Esquerdo
- 2º- Posiciona Rolo Inferior
- 3º- Posiciona Rolo Direito
- 4º- Posiciona Suportes Laterais
- 5º- Posiciona Suportes Verticais
- 6º- Posiciona Rotação

Imagem com 5 registos ou seja com 5 passagens o numero é possível ver na parte lateral direita onde surge o **MOD0 TEACH IN [5]**.




É possível editar os valores capturados bastos pressionar **BOTÃO LISTAR** <LISTAR>.



Para alterar um valor basta seleccionar uma linha que todos os valores da passagem correspondente são transferidos para a coluna da direita. Pressionando no Botão N° e introduzindo o numero da passagem.

Na coluna da direita ao pressionar um dos botões referentes aos rolos surgirá um teclado onde colocamos um novo valor, ao pressionar no botão **OK** vai incrementado a referencia do rolo, se não queremos alterar o valor basta pressionar **OK**.

A actualização valores totais só são alterados no fim do ciclo do teclado ou seja pressionar sempre **OK** até sair o teclado, nesse instante será actualizado o valor na tabela e se pressionarmos no SAIR saímos do modo **EDITAR**, se voltamos a seleccionar uma linha voltamos a correr todo o processo de alteração de uma passagem.


 <b>MARCOVIL, S.A</b> VISEU - PORTUGAL								<b>MCH 4R</b> <b>3000 / 50</b>	
Nº	Rolo 1E	Rolo 2E	Rolo 3D	Rolo 4D	Rolo 5I	Rolo 6I	Rotação	Nº	
1	120.8	120.8	70.4	70.4	30.0	30.0	0.0	R 1E	80.0
2	120.8	120.8	120.8	120.8	10.0	10.0	-100.0	R 2E	80.0
3	80.0	80.0	120.8	120.8	10.0	10.0	300.0	R 3D	120.8
4	120.0	120.0	70.0	70.0	10.0	10.0	1700.0	R 4D	120.8
5	130.0	130.0	130.0	130.0	30.0	30.0	0.0	R 5I	10.0
								R 6I	10.0
								ROT	300.0

ROLO 2E		
0	1	2
3	4	5
6	7	8
9	+/-	.
DEL	Back	OK
80.0		

### ▪ ALTERAÇÃO DE UM VALOR REGISTADO

Selecciona-se a linha onde se pretende alterar o registo, neste exemplo estamos a alterar o valor do registo do **ROLO 2E** da passagem 3 e pressiona-se o **BOTÃO R 2E**.

 <b>MARCOVIL, S.A</b> VISEU - PORTUGAL								<b>MCH 4R</b> <b>3000 / 50</b>	
Nº	Rolo 1E	Rolo 2E	Rolo 3D	Rolo 4D	Rolo 5I	Rolo 6I	Rotação	Nº	
1	120.8	120.8	70.4	70.4	30.0	30.0	0.0	R 1E	80.0
2	120.8	120.8	120.8	120.8	10.0	10.0	-100.0	R 2E	80.0
3	80.0	80.0	120.8	120.8	10.0	10.0	300.0	R 3D	120.8
4	120.0	120.0	70.0	70.0	10.0	10.0	1700.0	R 4D	120.8
5	130.0	130.0	130.0	130.0	30.0	30.0	0.0	R 5I	10.0
								R 6I	10.0
								ROT	300.0

EXIT		
------	--	--

Nº	Rolo 1E	Rolo 2E	Rolo 3D	Rolo 4D	Rolo 5I	Rolo 6I	Rotação
1	120.8	120.8	70.4	70.4	30.0	30.0	0.0
2	120.8	120.8	120.8	120.8	10.0	10.0	-100.0
3	80.0	80.0	120.8	120.8	10.0	10.0	300.0
4	120.0	120.0	70.0	70.0	10.0	10.0	1700.0
5	130.0	130.0	130.0	130.0	30.0	30.0	0.0

**MCH 4R 3000 / 50**

Nº 3

R 1E 80.0  
R 2E 80.0  
R 3D 120.8  
R 4D 120.8  
R 5I 10.0  
R 6I 10.0  
ROT 300.0

**ROLO 2E**

0 1 2  
3 4 5  
6 7 8  
9 +/- .  
DEL BKsp OK  
80.0

No fim de introduzir o valor correcto pressionamos o **BOTÃO OK**, e o valor foi actualizado como podemos ver na imagem.

Nº	Rolo 1E	Rolo 2E	Rolo 3D	Rolo 4D	Rolo 5I	Rolo 6I	Rotação
1	120.8	120.8	70.4	70.4	30.0	30.0	0.0
2	120.8	120.8	120.8	120.8	10.0	10.0	-100.0
3	80.0	80.0	120.8	120.8	10.0	10.0	300.0
4	120.0	120.0	70.0	70.0	10.0	10.0	1700.0
5	130.0	130.0	130.0	130.0	30.0	30.0	0.0

**MCH 4R 3000 / 50**

Nº 3

R 1E 80.0  
R 2E 85.5  
R 3D 120.8  
R 4D 120.8  
R 5I 10.0  
R 6I 10.0  
ROT 300.0

**ROLO 2E**

0 1 2  
3 4 5  
6 7 8  
9 +/- .  
DEL BKsp OK  
85.5

Se for necessário alterar mais valores na mesma passagem é só pressionar o **BOTÃO OK** até surgir a referência ao registo que queremos alterar (ex. alterar o **ROLO 5I**)

**MARCOVIL, S.A.**  
VISEU - PORTUGAL

**MCH 4R**  
3000 / 50

Nº	Rolo 1E	Rolo 2E	Rolo 3D	Rolo 4D	Rolo 5I	Rolo 6I	Rotação
1	120.8	120.8	70.4	70.4	30.0	30.0	0.0
2	120.8	120.8	120.8	120.8	10.0	10.0	-100.0
3	80.0	85.5	120.8	120.8	10.0	10.0	300.0
4	120.0	120.0	70.0	70.0	10.0	10.0	1700.0
5	130.0	130.0	130.0	130.0	30.0	30.0	0.0

Nº 3

R 1E 80.0

R 2E 85.5

R 3D 120.8

R 4D 120.8

R 5I 10.0

R 6I 10.0

RDT 300.0

ROLO 4D

0 1 2

3 4 5

6 7 8

9 +/-

DEL MCH OK

120.8

**MARCOVIL, S.A.**  
VISEU - PORTUGAL

**MCH 4R**  
3000 / 50

Nº	Rolo 1E	Rolo 2E	Rolo 3D	Rolo 4D	Rolo 5I	Rolo 6I	Rotação
1	120.8	120.8	70.4	70.4	30.0	30.0	0.0
2	120.8	120.8	120.8	120.8	10.0	10.0	-100.0
3	80.0	85.5	120.8	120.8	10.0	10.0	300.0
4	120.0	120.0	70.0	70.0	10.0	10.0	1700.0
5	130.0	130.0	130.0	130.0	30.0	30.0	0.0

Nº 3

R 1E 80.0

R 2E 85.5

R 3D 120.8

R 4D 120.8

R 5I 10.0

R 6I 10.0

RDT 300.0

ROLO 3D

0 1 2

3 4 5

6 7 8

9 +/-

DEL MCH OK

120.8

**MARCOVIL, S.A.**  
VISEU - PORTUGAL

**MCH 4R**  
3000 / 50

Nº	Rolo 1E	Rolo 2E	Rolo 3D	Rolo 4D	Rolo 5I	Rolo 6I	Rotação
1	120.8	120.8	70.4	70.4	30.0	30.0	0.0
2	120.8	120.8	120.8	120.8	10.0	10.0	-100.0
3	80.0	85.5	120.8	120.8	10.0	10.0	300.0
4	120.0	120.0	70.0	70.0	10.0	10.0	1700.0
5	130.0	130.0	130.0	130.0	30.0	30.0	0.0

Nº 3

R 1E 80.0

R 2E 85.5

R 3D 120.8

R 4D 120.8

R 5I 10.0

R 6I 10.0

RDT 300.0

ROLO 5

0 1 2

3 4 5

6 7 8

9 +/-

DEL MCH OK

11.8

**MARCOVIL, S.A.**  
VISEU - PORTUGAL

**MCH 4R**  
3000 / 50

Nº	Rolo 1E	Rolo 2E	Rolo 3D	Rolo 4D	Rolo 5I	Rolo 6I	Rotação
1	120.8	120.8	70.4	70.4	30.0	30.0	0.0
2	120.8	120.8	120.8	120.8	10.0	10.0	-100.0
3	80.0	85.5	120.8	120.8	11.8	10.0	300.0
4	120.0	120.0	70.0	70.0	10.0	10.0	1700.0
5	130.0	130.0	130.0	130.0	30.0	30.0	0.0

Nº 3

R 1E 80.0

R 2E 85.5

R 3D 120.8

R 4D 120.8

R 5I 11.8

R 6I 10.0

RDT 300.0

ROLO 6

0 1 2

3 4 5

6 7 8

9 +/-

DEL MCH OK

10.0

Continuando a pressionar o BOTÃO OK quando chegar ao ultimo registo da passagem o teclado sairá do ecrã, para alterar novo valor basta seleccioner a linha correspondente à passagem e pressionar o BOTÃO referente ao registo ex. R 3D e surgira o teclado onde colocar o novo valor.

[illegible]

Para **SAIR** do **MENU EDITAR** é necessário pressionar o **BOTÃO OK** até sair o teclado e depois pressionar no **BOTÃO EXIT** e neste momento voltamos ao **MENU** principal onde podemos continuar a captar valores ou ir guardar o programa ou executar o programa em Automático.

- **GUARDAR PROGRAMA TEACH IN**

Estando o modo **TEACH IN** activo é possível guardar um programa.

Para guardar um programa basta no menu **PRINCIPAL** activar o modo **TEACH IN**, surge a cor de laranja no lado direito a mensagem **TEACH IN [ON]**, de seguida pressionar **BOTÃO FICHEIRO**.

No menu FICHEIRO escolher:

**1º DESTINO:**

- DISCO MAQUINA – (ativo BOTÃO DISCO MAQUINA cor verde)
- DISCO FLASH – (ativo BOTÃO DISCO FLASH cor verde)

(Basta pressionar num dos dois BOTÕES)

## 2ª OPERAÇÃO A REALIZAR



- ABRIR – Pressionar BOTÃO ABRIR
- GUARDAR – Pressionar BOTÃO GUARDAR
- APAGAR – Pressionar BOTÃO APAGAR

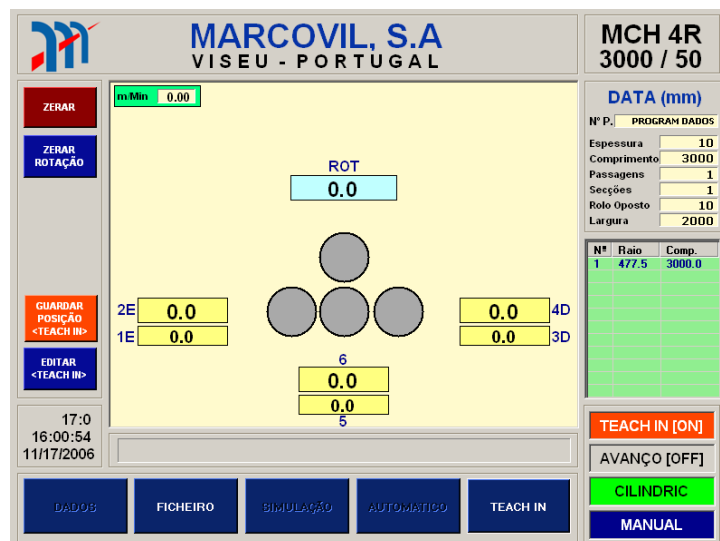
### 3º DAR NOME DO PROGRAMA

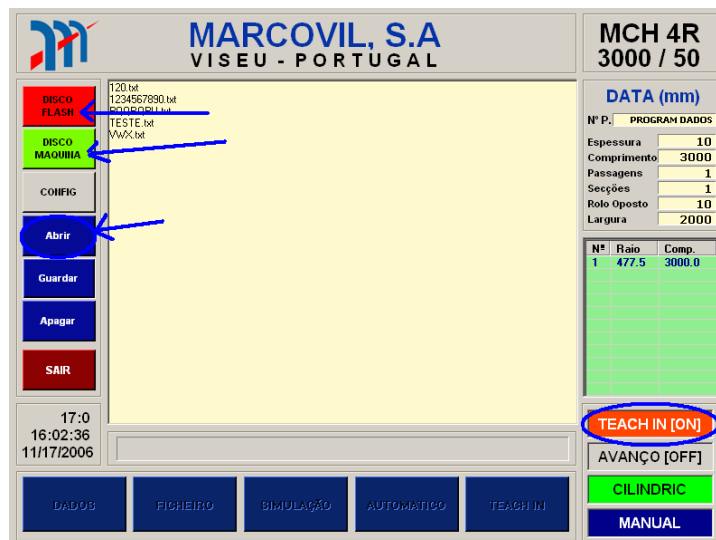
Nome sem extensão ex. *TESTE.txt* colocar na caixa de texto do teclado do lado direito *TESTE* e pressionar o *BOTÃO OK*

#### ▪ ABRIR PROGRAMA TEACH IN

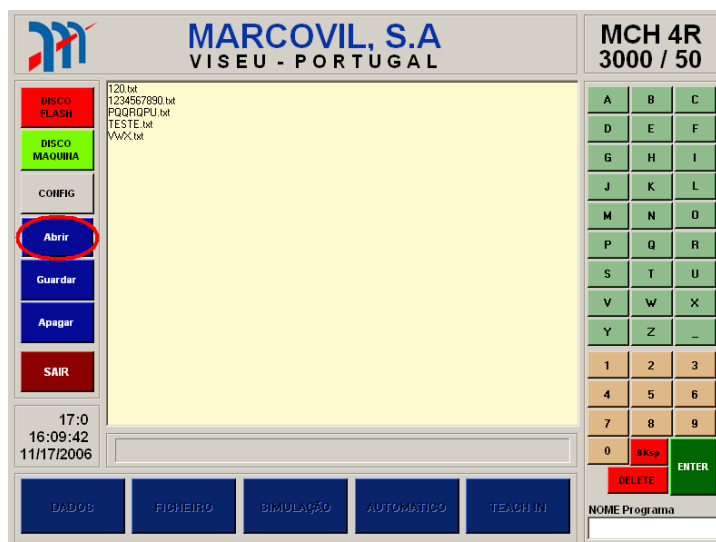
Estando o modo *TEACH IN* activo é possível abrir um programa guardado para ser editado ou executado no modo *AUTOMATICO*.

Para abrir um programa guardado basta no menu *PRINCIPAL* activar o modo *TEACH IN*, surge a cor de laranja no lado direito a mensagem *TEACH IN [ON]*, de seguida pressionar *BOTÃO FICHEIRO*.

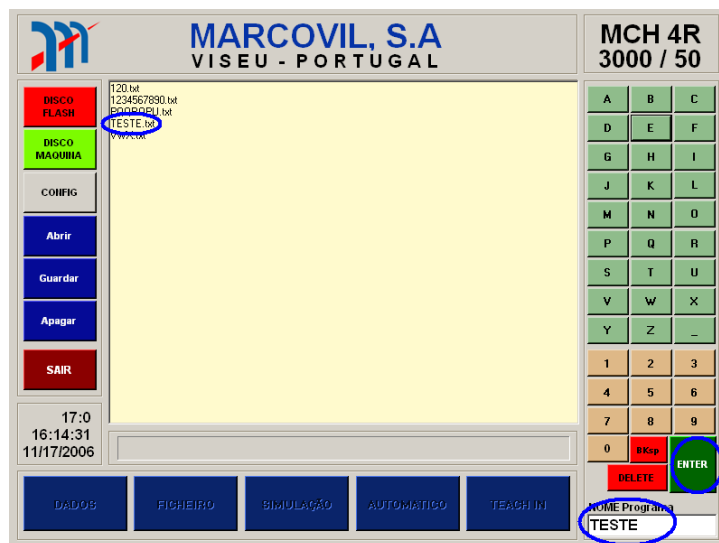




É possível escolher se o programa está num disco flash externo ou esta na máquina, para essa selecção basta pressionar nos dois primeiros **BOTÕES DISCO (FLASH, MAQUINA)**, o que estiver a verde está activo. O próximo passo é pressionar na função que pretende efectuar, o caso é **ABRIR** pressiona **BOTÃO ABRIR**.



Surge um teclado do lado direito onde vamos colocar o nome do programa sem a extensão. Ex. **TESTE.txt** coloca-mos **TESTE**



Pressionar **ENTER** para abrir o programa e de seguida pressionar **SAIR** para sair para o menu **PRINCIPAL**.

No menu **PRINCIPAL** é possível alterar o programa pressionado no **BOTÃO EDITAR**, ou passar á execução do programa pressionando o **BOTÃO AUTOMÁTICO**.

## **IX. CALANDRAGEM DE PEÇAS CÔNICAS**

A Calandragem de cones não está prevista nos sistemas *CNC*, devido à situação de escorregamento da chapa. Este escorregamento é consequência da diferença de velocidades periféricas por causa das diferenças de diâmetros a calandrar. Escorregamento esse que altera de forma aleatória tornando a sua correcção automática inviável, através da leitura do movimento efectivo da periferia dos rolos, na sua rotação. Como a velocidade da chapa é diferente da velocidade periférica dos rolos, o seu posicionamento linear não é possível, falhando qualquer sistema computadorizado de controlo numérico.

A calandragem cónica, é pois conseguida no modo **Manual**, obedecendo a regras de pre-planificação da chapa - consultar manual de instruções da máquina cap. 3.5 CALANDRAGEM CÔNICA -, e com a experiência do operador da máquina.

A calandragem cónica é facilmente conseguida nas máquinas da série MCH 4R, devido ao batente de cones e à facilidade de inclinação dos rolos para o efeito.

A *Marcovil*, S.A. está a pesquisar a possibilidade de adaptação do *CNC Marcovil* à calandragem de cones, afim de oferecer ao seu clientes a resolução dos problemas de calandragem cónica, num futuro próximo.

## X. PROGRAMAÇÃO DE VIROLAS CILÍNDRICAS

A programação de virolas de tubos completamente cilíndricos é regida pela mesma estrutura atrás definida, mas tendo em conta que o número de **SECÇÕES** será sempre zero.

Depois de introduzir o valor zero na caixa de escrita **SECÇÕES** pressionar o **BOTÃO OK** do teclado. Outro sub menu surge e o cursor posiciona-se na caixa de escrita **LONGITUD**. o valor a introduzir agora terá de ser igual ao valor do comprimento da chapa introduzido no início da programação. O **CNC** assume ( depois de pressionado a **BOTÃO OK DO TECLADO**) e regista os dois valores iguais: comprimento da chapa e comprimento da Secção e automaticamente apresenta na caixa de valores **RAIO** um valor equivalente ao raio de um circulo com o perímetro igual aos valores introduzidos. Aceitar esse valor pressionando a **BOTÃO DE OK**. A mensagem **DADOS OK** aparece no fim do ecrã. O programa está registado e poderá ser executado da forma já descrita.

Estas instruções são tidas como sucintas e principais, cobrindo no entanto o essencial de todas as formas que se pretendam calandrar. A **Marcovil** - Metalomecânica de Viseu S.A. reserva-se no direito de modificar, alterar ou reversionar, quando necessário, o texto ou conteúdo destas instruções, sem aviso prévio.





MARCOVIL,S.A.

Apartado

Empresa / Cliente	Marcovil,S.A
Cliente	
Maquina	
Descrição do projecto	Calandra Hidraulica 4R2500/26
Número de Obra	1439
Número de Registo	

Fabricante (Empresa)	MARCOVIL,S.A.
Nome do projecto	CNC4R250026
Realizar	
Versão Software	
Localização de instalação	+ET1
Responsável pelo projecto	
Características da peça	COMANDO CNC

Criado em	21-01-2008
-----------	------------

Editado em	30-05-2008	de (abreviatura) Paulo
------------	------------	------------------------

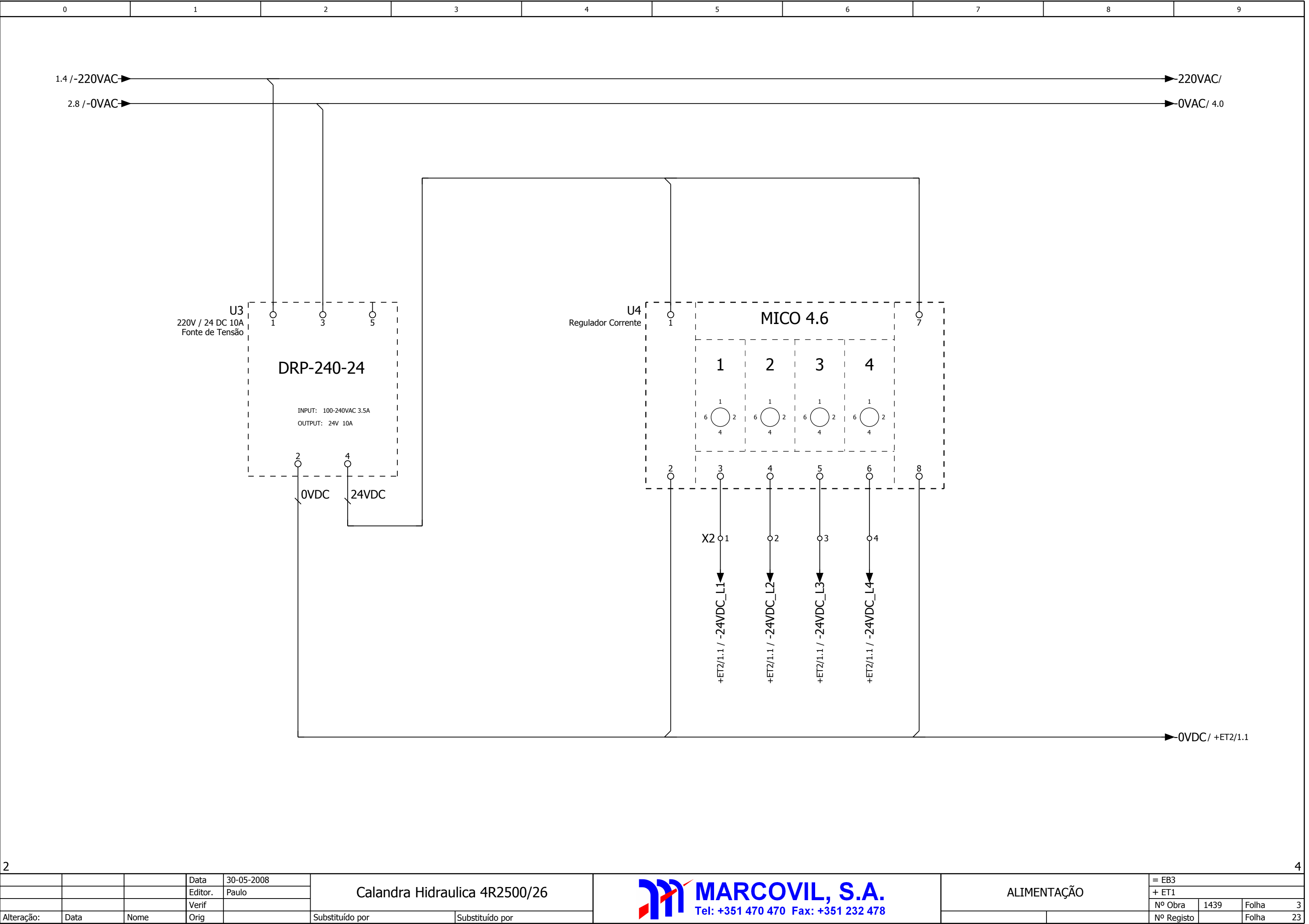
Quantidade de páginas	23
-----------------------	----

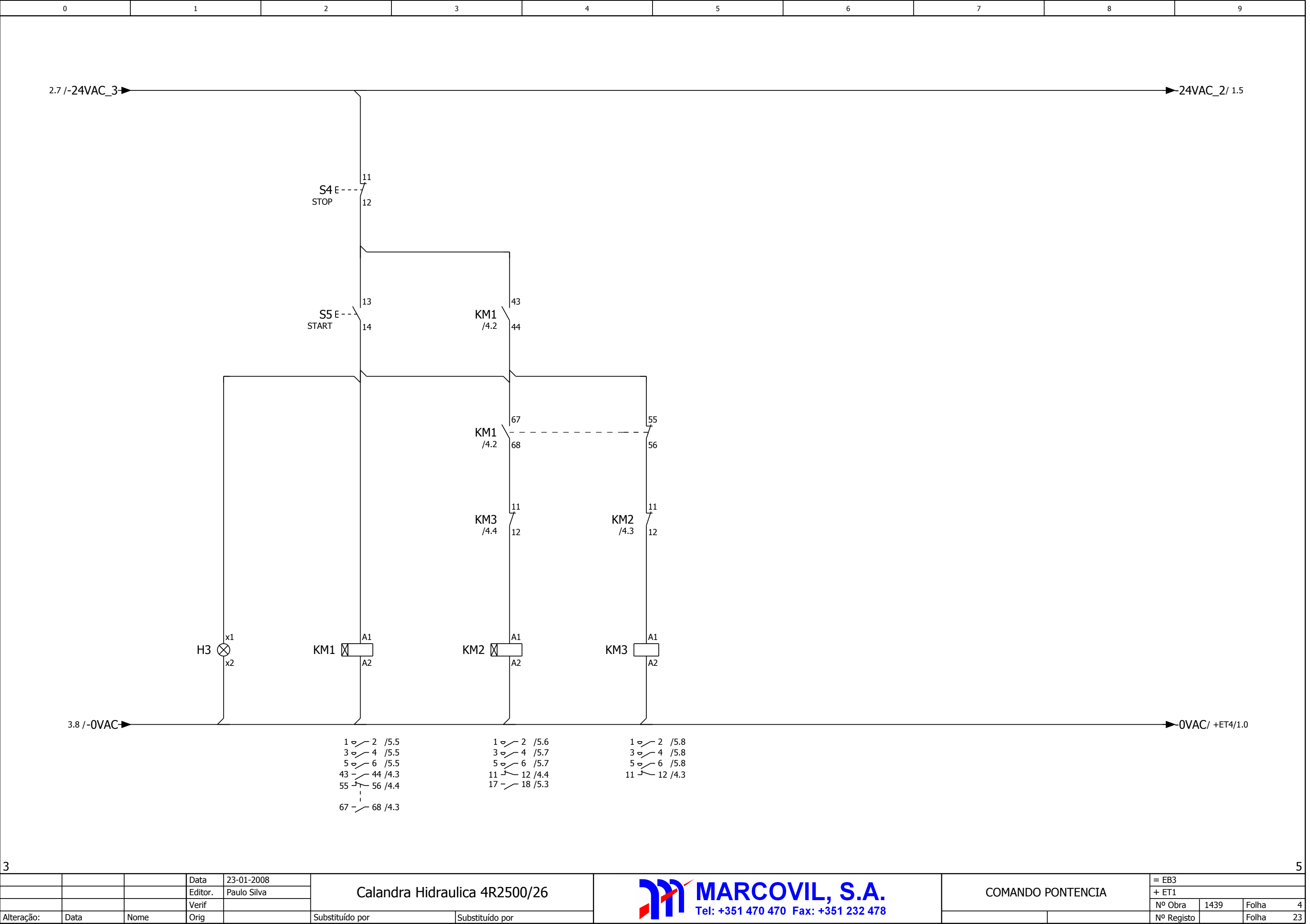




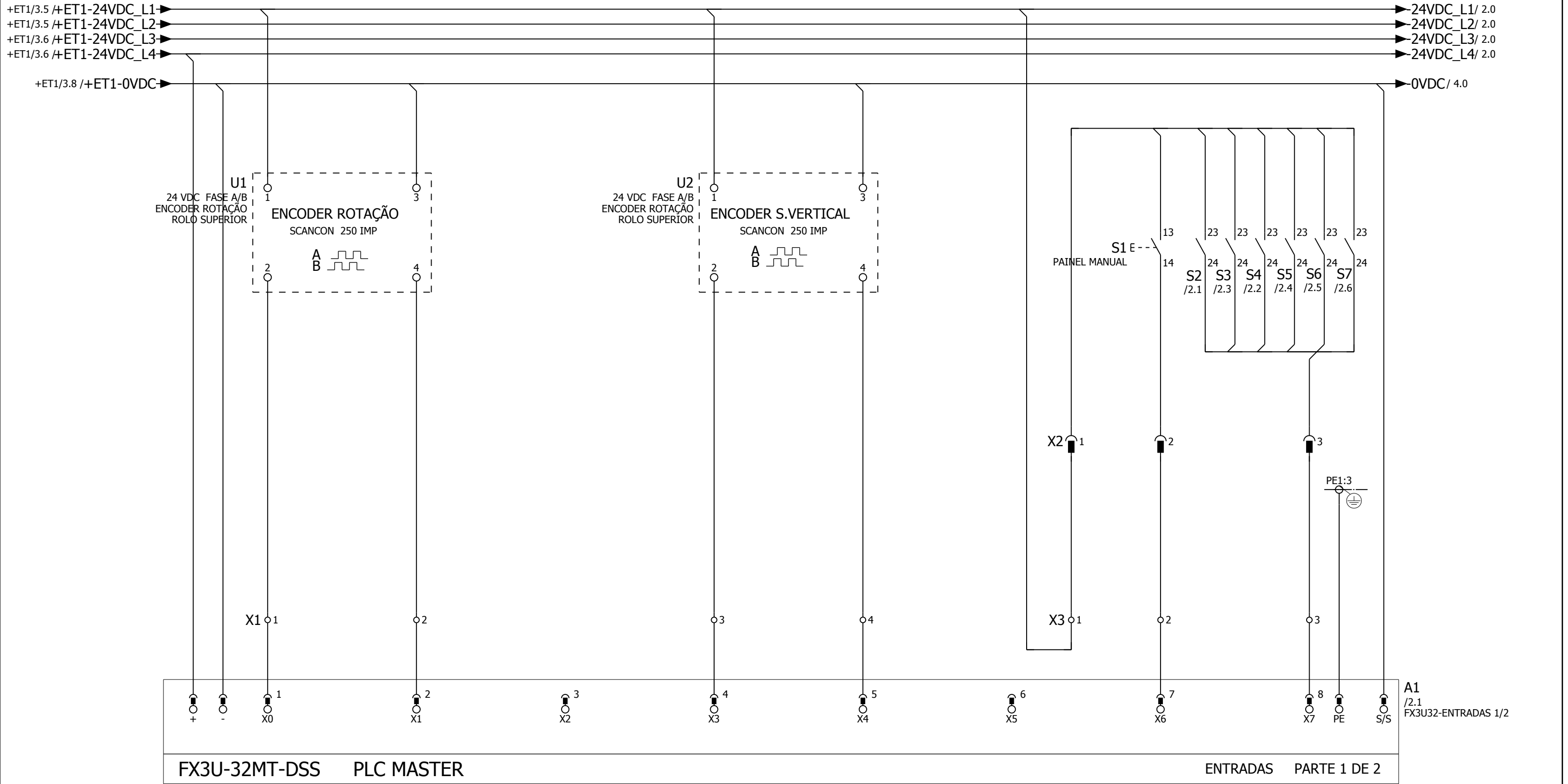


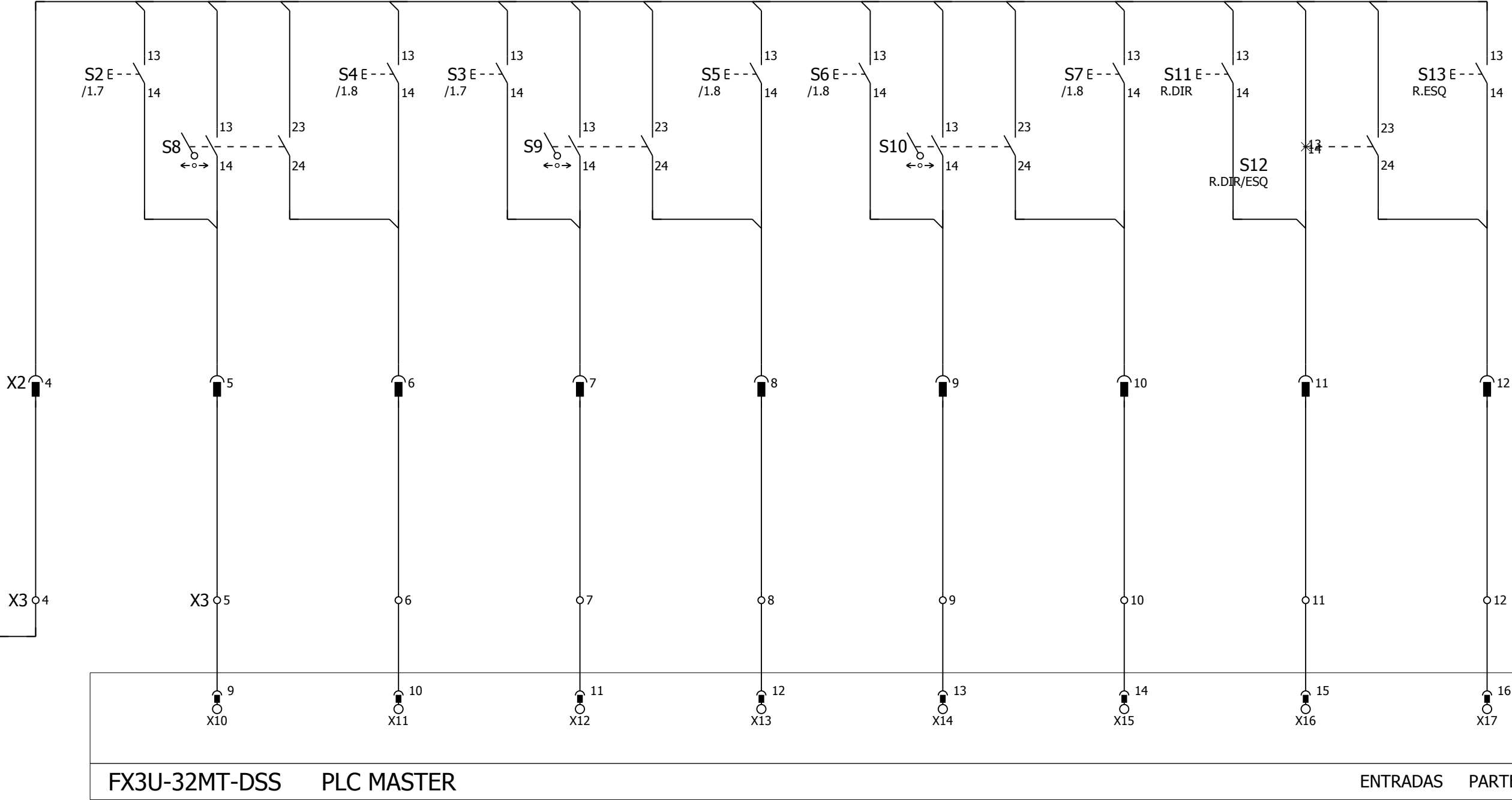
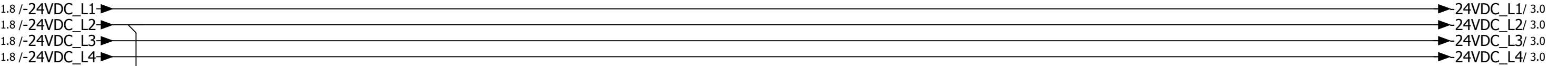
7.3.0\_05\_a











Sobe Rolo Lateral Esquerdo

Desce Rolo Lateral Esquerdo

Sobe Rolo Lateral Direito

Desce Rolo Lateral Direito

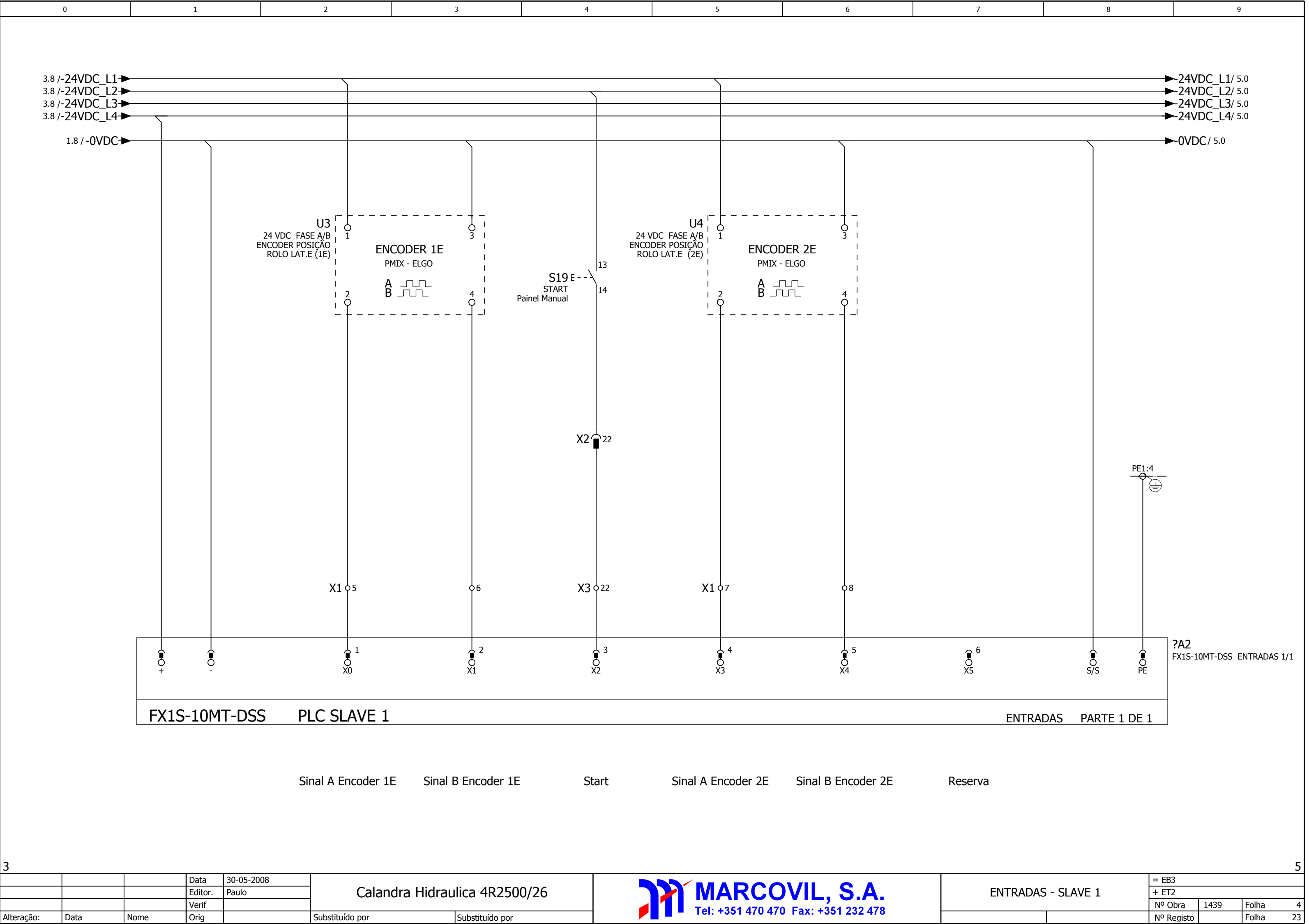
Sobe Rolo Inferior

Desce Rolo Inferior

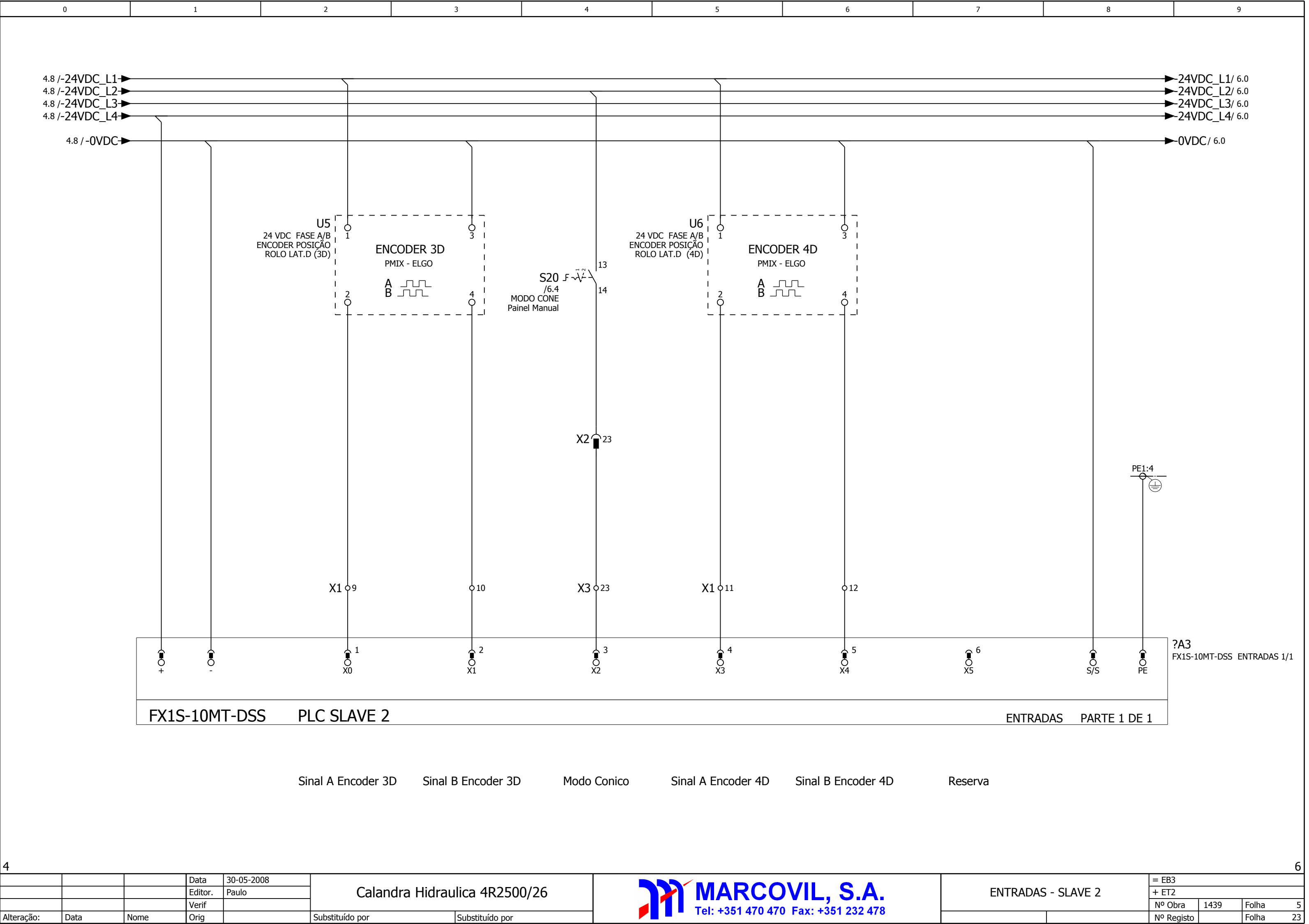
Rotação Para Direita

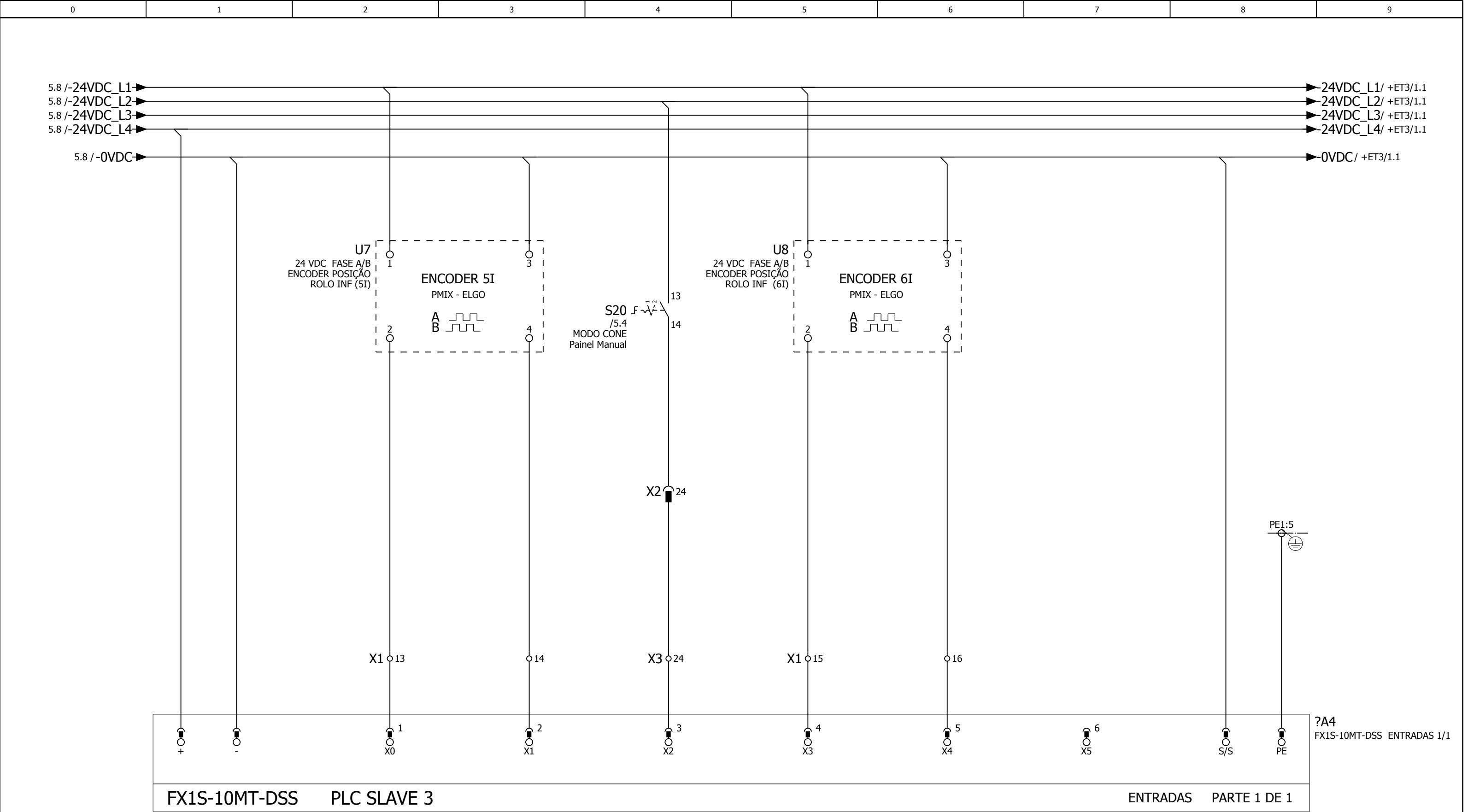
Rotação Para Esquerda











Sinal A Encoder 5I

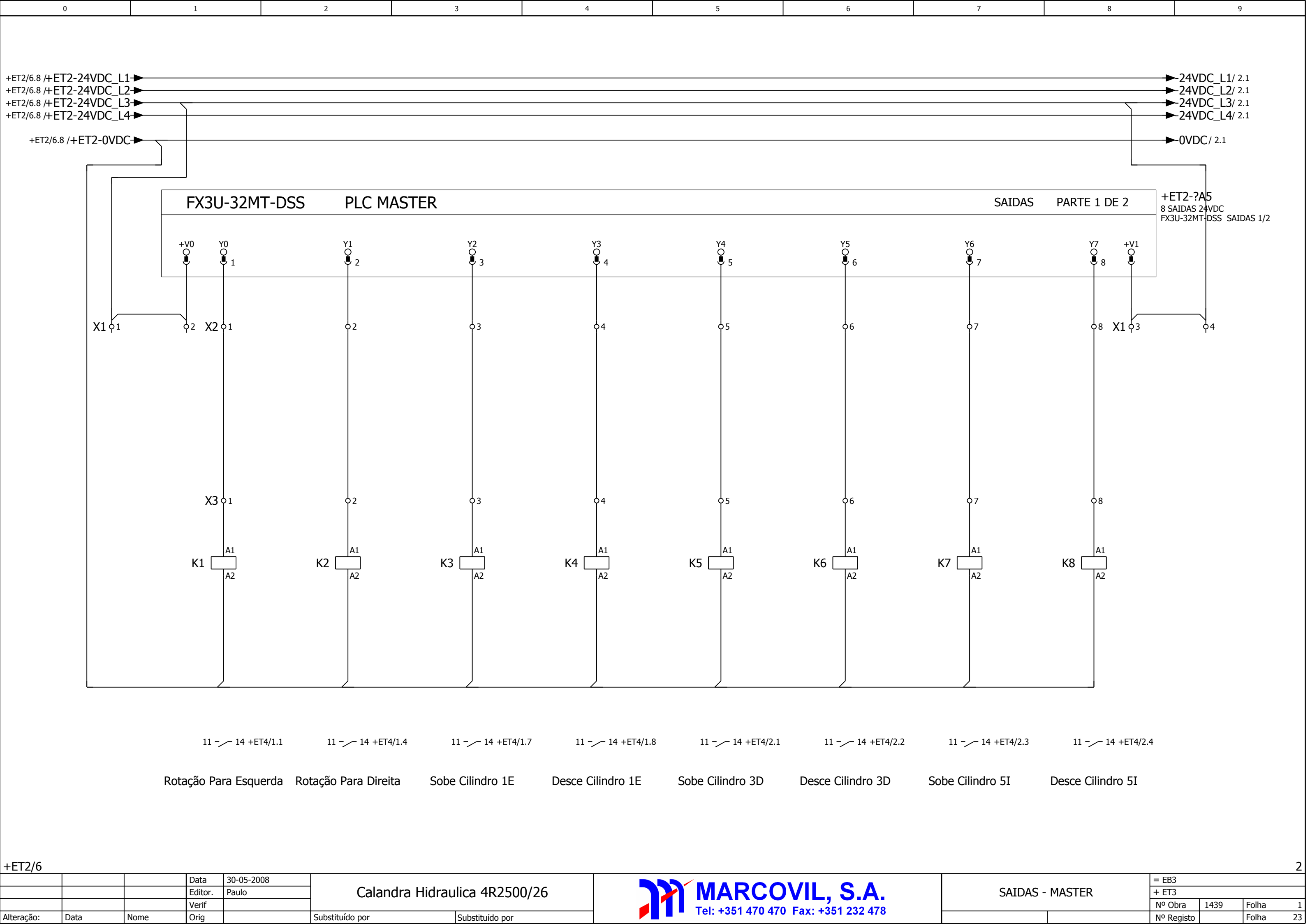
Sinal B Encoder 5I

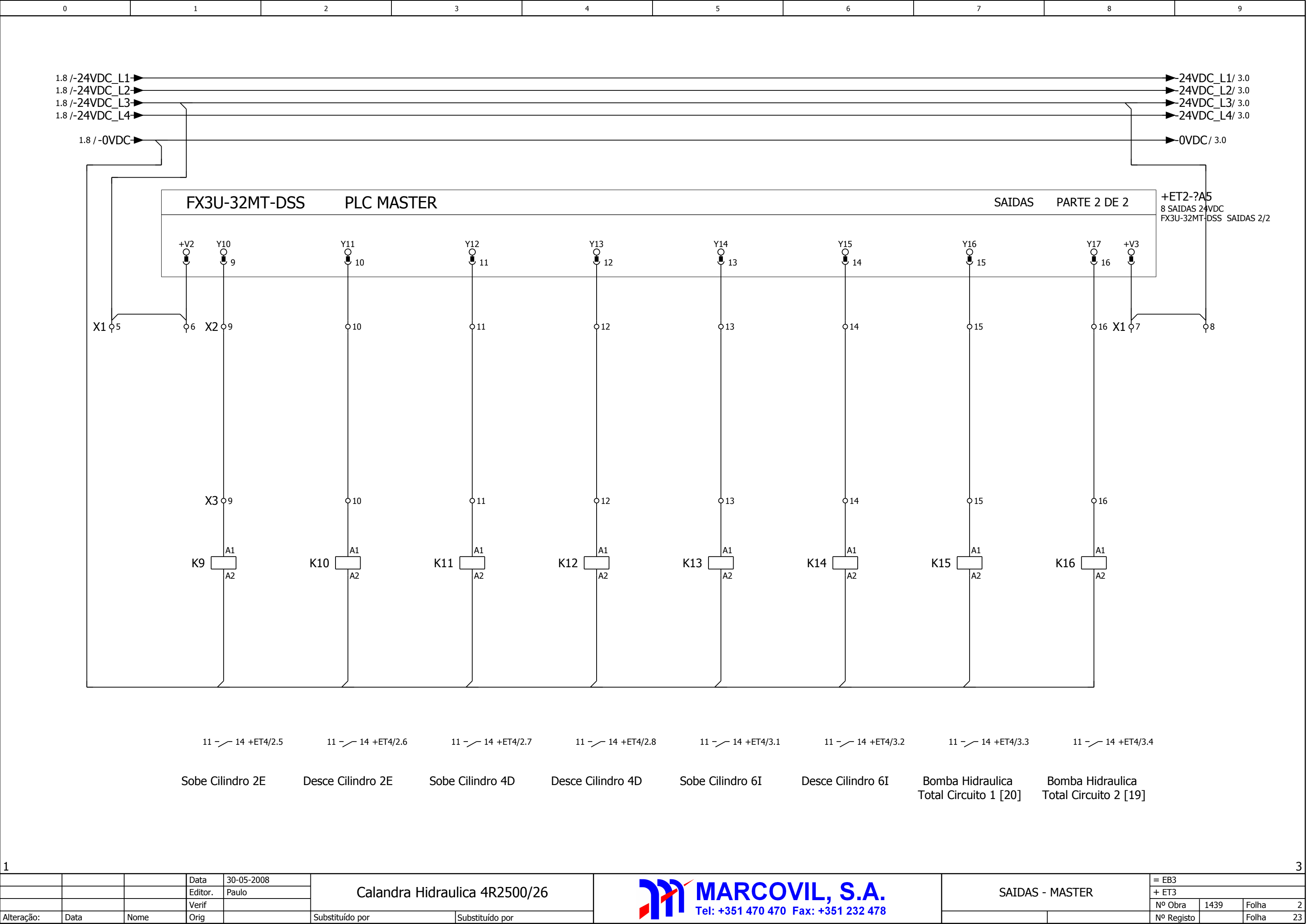
Reserva

Sinal A Encoder 6I

Sinal B Encoder 6I

Reserva



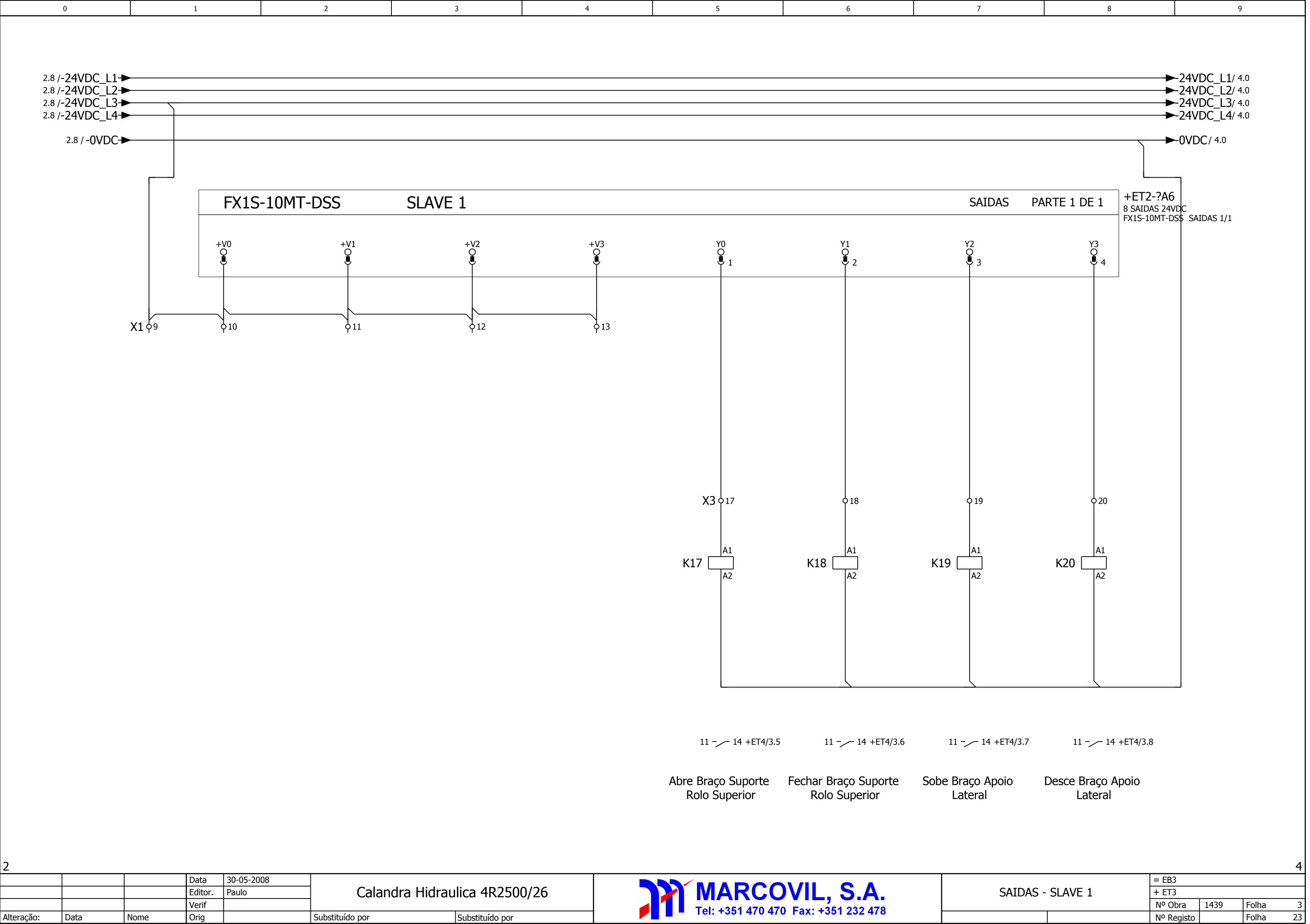


1

3

			Data	30-05-2008	Calandra Hidraulica 4R2500/26		 <b>MARCovil, S.A.</b> Tel: +351 470 470 Fax: +351 232 478	SAIDAS - MASTER			= EB3		
			Editor.	Paulo							+ ET3		
			Verif								Nº Obra	1439	Folha
Alteração:	Data	Nome	Orig		Substituído por	Substituído por				Nº Registo		Folha	23

7.3.0\_05\_a

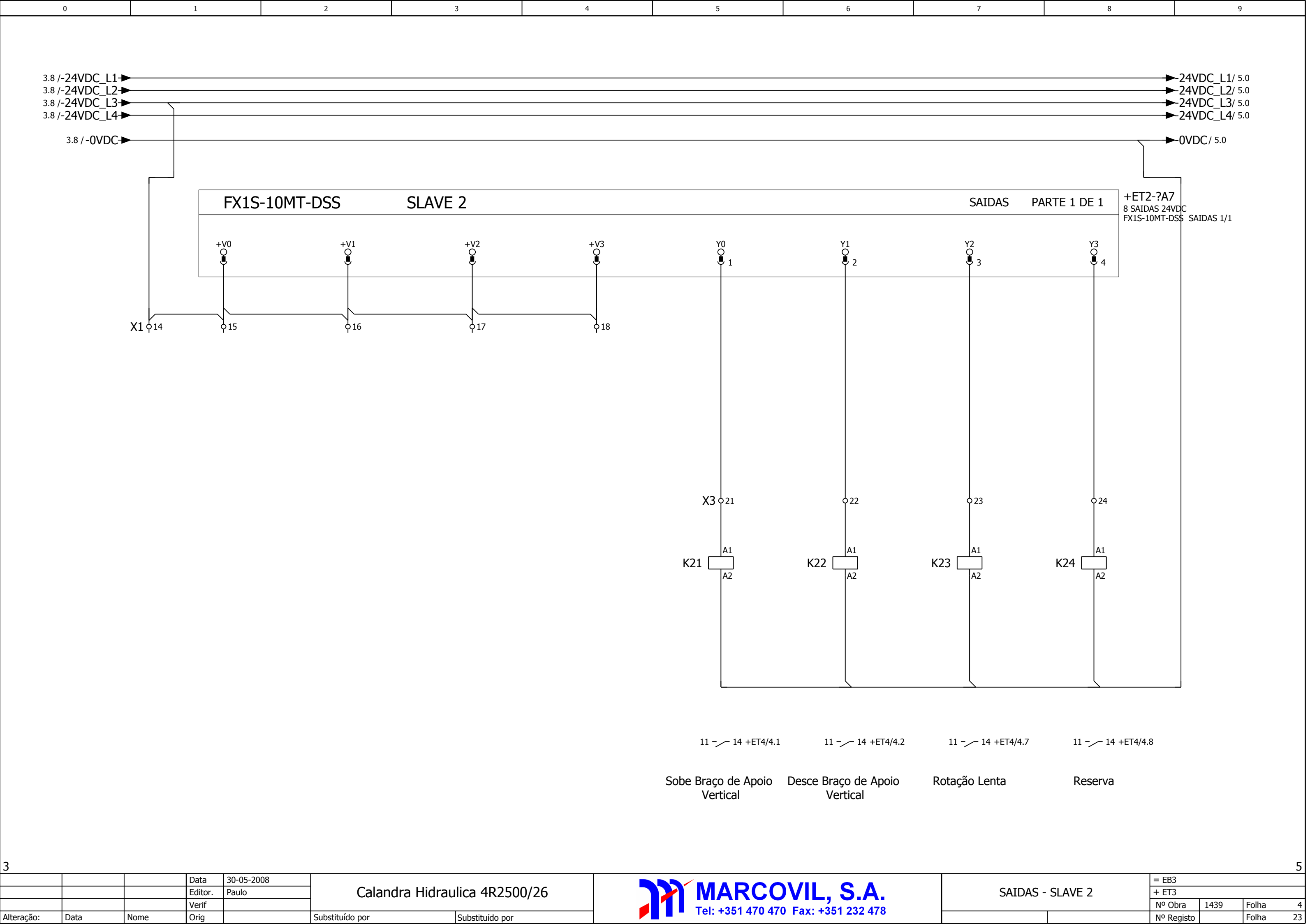


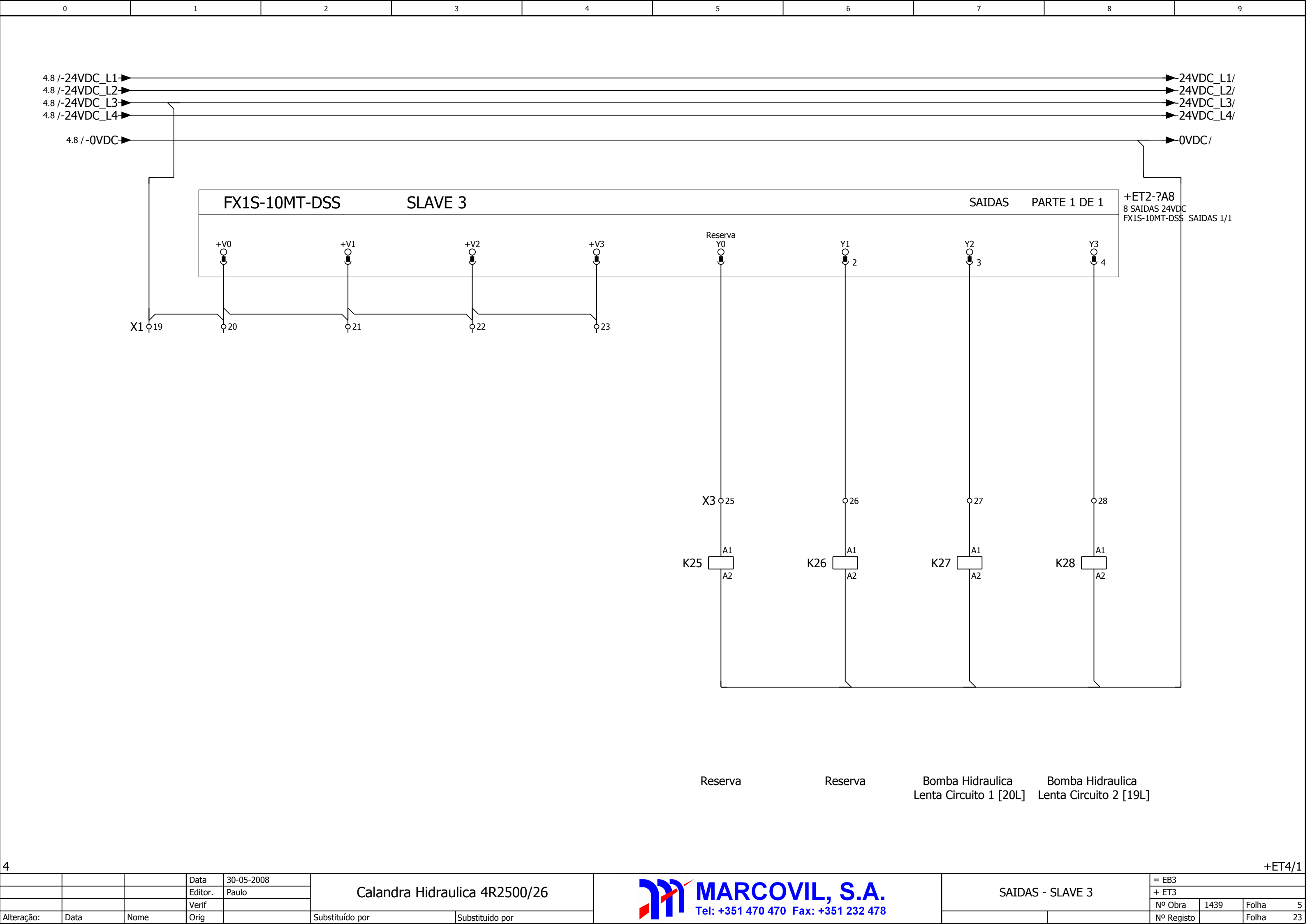
2

4

			Data	30-05-2008	Calandra Hidraulica 4R2500/26		 <b>MARCovil, S.A.</b> Tel: +351 470 470 Fax: +351 232 478	SAIDAS - SLAVE 1			= EB3		
			Editor.	Paulo				+ ET3					
			Verif										
Alteração:	Data	Nome	Orig		Substituído por	Substituído por				Nº Obra	1439	Folha	3
										Nº Registo		Folha	23

7.3.0\_05\_a



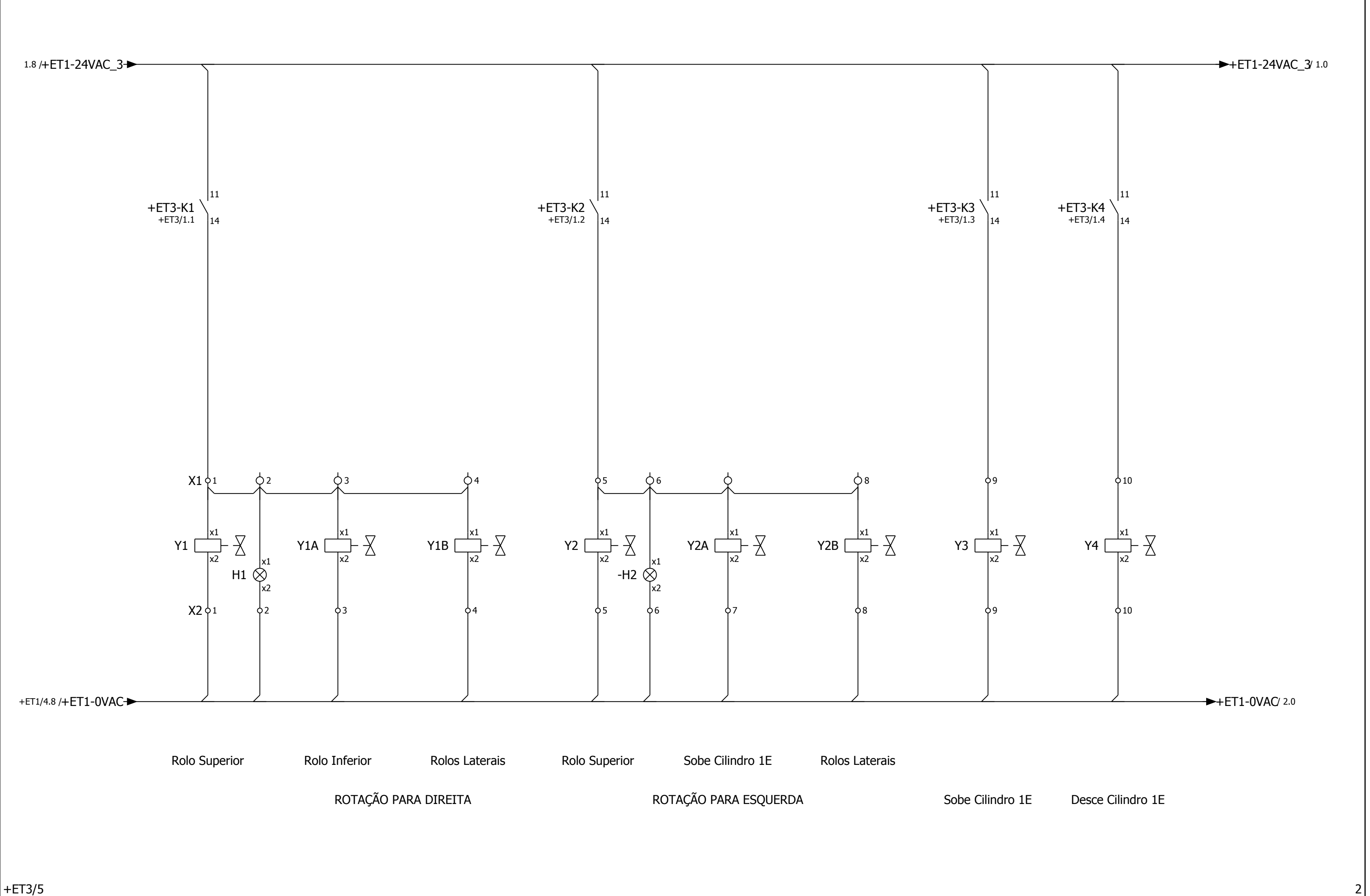


4

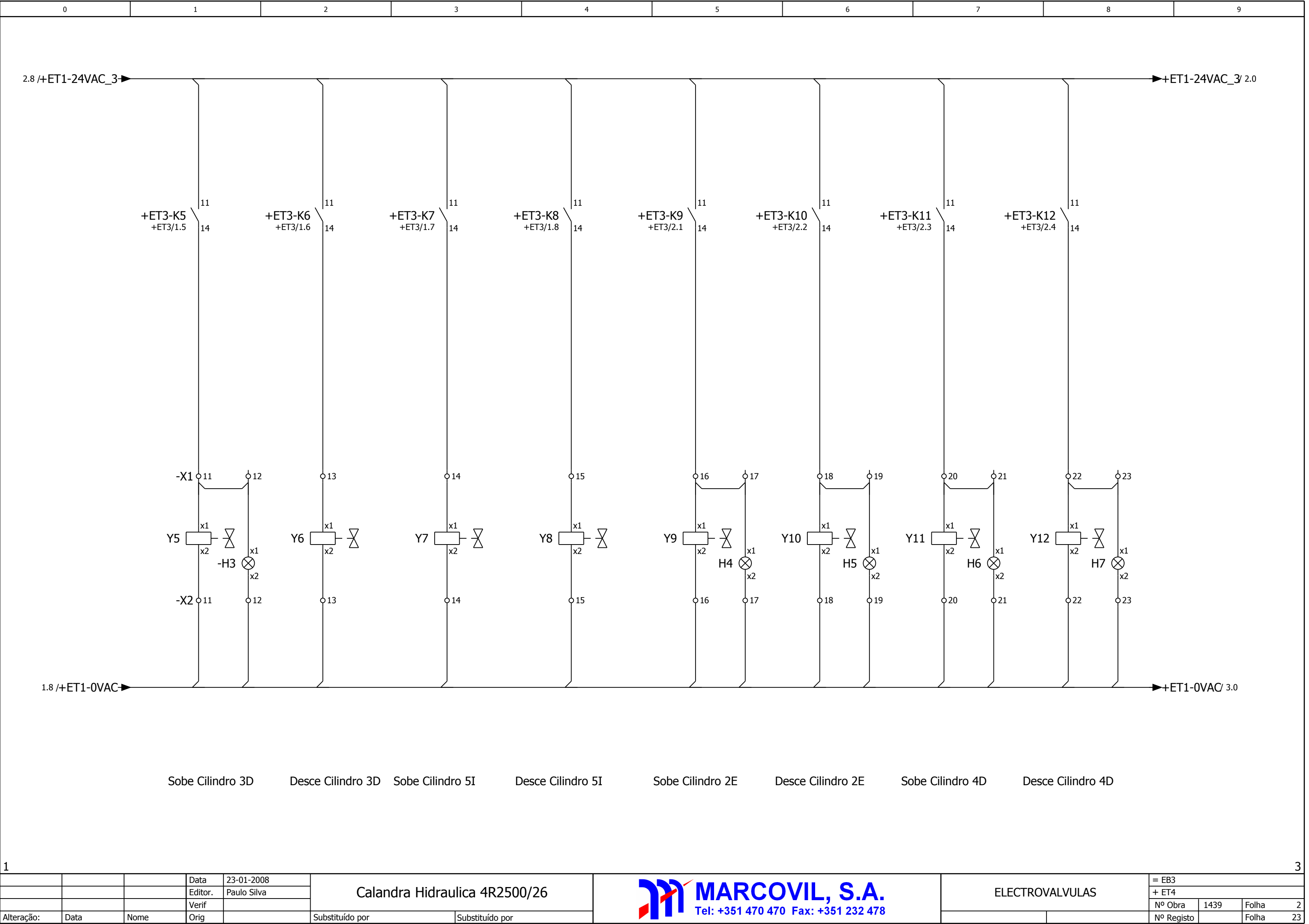
+ET4/1

			Data	30-05-2008	Calandra Hidraulica 4R2500/26		 <div>MARCOVIL, S.A. Tel: +351 470 470 Fax: +351 232 478</div>	SAIDAS - SLAVE 3		= EB3		
			Editor.	Paulo						+ ET3		
			Verif					Nº Obra	1439	Folha	5	
Alteração:	Data	Nome	Orig		Substituído por				Nº Registo		Folha	23

7.3.0\_05\_a







15

15

Y8

x1  
x2

15

15

16

16

17

17

Y9

x1  
x2

H4

x1  
x2

16

17

18

18

19

19

Y10

x1  
x2

H5

x1  
x2

18

19

20

20

21

21

Y11

x1  
x2

H6

x1  
x2

20

21

22

22

23

23

Y12

x1  
x2

H7

x1  
x2

22

23

1.8 /+ET1-0VAC

+ET1-0VAC/ 3.0

Sobe Cilindro 3D

Desce Cilindro 3D

Sobe Cilindro 5I

Desce Cilindro 5I

Sobe Cilindro 2E

Desce Cilindro 2E

Sobe Cilindro 4D

Desce Cilindro 4D

1

3

7.3.0\_05\_a







